UNIVERSITY of PENNSYLVANIA

PHILADELPHIA, PENNSYLVANIA 19104

223

UTILITY OF AUTOMATIC CLASSIFICATION SYSTEMS FOR INFORMATION STORAGE AND RETRIEVAL

Barry Litofsky

Presented to the Faculty of the Graduate School of Arts and Sciences of the University of Pennsylvania in partial fulfillment of the requirements for Doctor of Philosophy.

May 1969

The research described in the dissertation by Dr. B. Litofsky has been partially supported by Bell Telephone Laboratories, Holmdel, New Jersey, which supported Dr. Litofsky throughout the research and provided IBM 360/65 computer time, and partially by The Moore School of Electrical Engineering, University of Pennsylvania, under contract NOnr 551(40) which provided dissertation research supervision as well as IBM 7040 and IBM 360/65 computer time.

ABSTRACT

UTILITY OF AUTOMATIC CLASSIFICATION SYSTEMS FOR INFORMATION STORAGE AND RETRIEVAL

by
Barry Litofsky

Supervised by Professors Noah S. Prywes
and David Lefkovitz

Large-scale, on-line information storage and retrieval systems pose numerous problems above those encountered by smaller systems. The more critical of these problems involve: degree of automation, flexibility, browsability, storage space, and retrieval time. A step toward the solution of these problems is presented here along with several demonstrations of feasibility and advantages.

The methodology on which this solution is based is that of a posteriori automatic classification of the document collection. Feasibility is demonstrated by automatically classifying a file of 50,000 document descriptions. The advantages of automatic classification are demonstrated by establishing methods for measuring the quality of classification systems and applying these measures to a number of different classification strategies. By indexing the 50,000 documents by two independent methods, one manual and one automatic, it is shown that these advantages are not dependent upon the indexing method used.

It was found that among those automatic classifi-

cation algorithms studied, one particular algorithm, CLASFY, consistently outperformed the others. In addition, it was found that this algorithm produced classifications at least as good, with respect to the measures established in this dissertation, as the a priori, manual classification system currently in use with the aforementioned file.

The actual classification schedules produced by CLASFY in classifying a file of almost 50,000 document descriptions into 265 categories are included as an appendix to this dissertation.

This dissertation is dedicated to my
wife, Francine, a literature chemist,
who feels that work such as this
will put her out of a job.

ACKNOWLEDGEMENTS

The author is indebted to a number of people who gave their advice, assistance, or encouragement.

To Drs. Noah S. Prywes and David Lefkovitz, the supervisors of this dissertation, the author expresses his gratitude for their assistance from start to finish, and especially to Dr. Prywes for the suggestion to undertake work along these lines.

The Bell Telephone Laboratories are acknowledged for the financial support and computation facilities which have been made available to the author during this project. In particular, Dr. W. B. Macurdy is sincerely thanked for his advice and encouragement throughout the author's graduate work.

Pauline Atherton and Robert R. Freeman of Syracuse University and the Center for Applied Linguistics, respectively, are acknowledged for their help in locating a data file suitable for this dissertation. Gratitude is expressed to the United States Atomic Energy Commission, Division of Technical Information Extension, for the use of Nuclear Science Abstracts data files and especially to Joel S. O'Connor, formerly of that organization for his aid in obtaining and understanding those files.

The author also thanks Gloria Smith of Lawrence Radiation Laboratories for supplying the "live" retrieval

requests which were applied to the above data files.

Additionally, gratitude is expressed to Thomas Angell and Don Headley of Computer Command and Control Company for their help in understanding and modifying the original computer programs for CLASFY and hierarchy generation.

A special note of acknowledgement is given to James R. McEowen whose patient ear and encouragement was appreciated by the author throughout the course of this work.

The author is everlastingly grateful to his parents for their instilling in him the importance of education and the sense of dedication to academic achievement which led to this dissertation and the doctoral degree which it signifies.

Last, but first in the author's heart, is his wife, Francine, who besides typing this dissertation, has provided the author infinite encouragement and joy and has endured many hardships during the course of his graduate study. Without her encouragement, this dissertation would not have been possible.

INDEX

	Page
Agglomerative Methods	45
Automata Theory	44
Browsing	27ff,39,125,133
Buckets	15
Cells	15,34,80,102,145ff
Cells Searched	53,108,120ff
CLASFY	
Algorithm	58ff
Input Order	78,115,148
Present Experiments	51,52
Previous Experiments	50,51,52
Results of Classification	110ff,152ff
Classification Time	43,44,57,75,153,154
Colon Classification	38
Computer Memories	12,21ff
Clumping	42ff
Clustering	42ff
Data Files	105,106,157ff
Dewey Decimal Classification	24,25,38
Directory	12,13,19ff,29ff
Discriminant Method	46,49
Divisive Methods	45,50
Document Retrieval	7ff
Document Storage	4ff

	Page
Documents Retrieved	106,183ff
Documents Searched	53,108,120ff
Examples of Classifications	67ff,133ff
Exclusive and Inclusive Classification	45
Factor Analysis	46
Forward Ordering	
Algorithm	86ff,92ff
Classification	87,98ff
Results of Classification	110ff
General Inquirer	169
Hierarchy Tree	27ff,79ff,142ff
Human Classification	103,104,110ff
Indexing	4,10,19,37,54ff,105,165ff
Information Explosion	1
Inverted Files	11ff,18,19,117ff
Key-to-Node Table	30,83ff,132ff
Keywords per Cell	15ff,53,107,110ff
KWIC	56
Latent Class Analysis	51
Linear Programming Theory	44
List-Organized Files	12,13
$\log_N N_d$	75,154
Measures of Classification	40ff,107ff
$N_d \log_N N_d$	44,57
Node-to-Key Table	82,83,138ff

	Page
Nuclear Science Abstracts	157ff
Numerical Taxonomy	54
On-Line Interaction	7,9,10,27ff,31ff
Pattern Recognition	54
Personal Information System	3
Project SMART	55,56
Random Classification	101,102,110ff
Retrieval Requests	106,116ff,179ff
Reverse Ordering	
Algorithm	91ff
Classification	87,98ff
Results of Classification	110ff
Sensitivity Factor, E	63,69,72,75ff
Serial Files	11,18,149ff
Stratification Number, N	59,69,73ff
Suffix Deletion	169ff
Surrogate Storage	5,6
Terminal Node Table	86,87
Universal Decimal Classification	19,38
Ward Grouping Program	47,48,50

TABLE OF CONTENTS

	Page
Acknowledgements	iii
Index	v
List of Figures	xii
List of Tables	xv
Bibliography	xvii
CHAPTER 1 INTRODUCTION TO INFORMATION STORAGE AND RETRIEVAL	
1.1 Magnitude of the Problem	1
1.2 IS&R Functions	3
1.2.1 Storage	4
1.2.2 Retrieval	7
1.3 Inadequacies of Current Systems	9
CHAPTER 2 CONCEPTS OF IS&R BASED ON AUTOMATIC CLASSIFICATION	
2.1 Classification Parameters	15
2.2 Advantages of A Posteriori, Hierarchical, Automatic Classification Systems for On-Line Retrieval Systems	19
2.2.1 Automatic Classification: Directory Size Reduction	19
2.2.2 Automatic, A Posteriori: Flexible	22
2.2.3 Hierarchical, On-Line: Browsable	27
2.2.4 Hierarchical: Further Directory Size Reduction	29
2.2.5 On-Line Retrievals: Memory Accesses Reduction	31
2.3 Contributions of the Dissertation	35

CHAPTER 3 LITERATURE REVIEW

3.1 Content of this Chapter 37

3.2 General Critique of Prior Experiments 37

3.3 Automatic Classification 45

3.4 Indexing and Automatic Indexing 54

CHAPTER 4 EXPERIMENTAL CLASSIFICATION STRATEGIES

4.1 Introduction 57

4.2 A Hierarchical Classification Algorithm (CLASFY) 58

4.2.1 Description of the Algorithm 58

4.2.2 Classification Example 67

4.2.3 Unusual Situations 70

4.2.4 Discussion of Parameters 73

4.2.4.1 Stratification Numbers 73

4.2.4.2 Sensitivity Factor 75

4.2.5 Ordering of Input 78

4.3 Hierarchy Generation 79

4.3.1 Node-to-Key Table 82

4.3.2 Key-to-Node Table 83

4.3.3 Terminal Node Table 86

4.4 Forward and Reverse Classifications 87

4.4.1 Forward Ordering 88

4.4.2 Reverse Ordering 91

4.4.3 Modified Orderings 92

4.4.4 Classification Algorithm 98

4.5 Random "Classification" 101

4.6 Human (A Priori) Classification 103

CHAPTER 5 EXPERIMENTS AND RESULTS

5.1 Data Files 105

5.2 Experimental Measures of Quality of Classification 107

5.3 Keywords per Cell 110

5.4 Results of Retrieval Requests 116

5.4.1 Theoretical Results with Inverted File 117

5.4.2 Cells Searched 120

5.4.3 Documents Searched 126

5.5 Quality of Hierarchy 132

5.5.1 Size of Key-to-Node Table 132

5.5.2 Subjective Evaluation and Example 133

5.5.3 Documents in Cells 145

5.6 Summary of Results 148

5.7 Bonus Result - Average Length of Search in Serial Files 149

CHAPTER 6 CONCLUSIONS AND SUGGESTION FOR FUTURE RESEARCH

6.1 General Conclusions 152

6.2 Future Research 152

APPENDIX A NUCLEAR SCIENCE ABSTRACTS DATA FILES

A.1 Source of the Data 157

A.2 Keyword Files 158

A.3 Entry (Title Word) File 163

A.3.1 Nature of the File 163

A.3.2 Semi-Automatic Indexing 165

A.4 File Statistics 172

APPENDIX B DOCUMENT RETRIEVALS

B.1 Retrieval Requests 179

B.2 Documents Retrieved 183

LIST OF FIGURES

Figure	Title	Page
1-1	Growth of Journals and Abstract Journals	2
1-2	Typical Storage of Documents	5
1-3	Retrieval of Documents	8
2-1	Classification Parameters	17
2-2	Sample Classifications	21
2-3	Dewey Decimal Classification Schedule	25
2-4	Inverted File on Cells	30
2-5	Inverted File on Nodes	30
4-1	Macro-flowchart of CLASFY	62
4-2	Part of Classification using CLASFY	66
4-3	A Sample File of Document Descriptions	68
4-4	Partition of Top Level, N = 2, E = 0	69
4-5	Classification Tree, 14 Documents, N = 2, E = 0	71
4-6	Effects of E on Keys per Cell, N = 3, N_d = 2254	77
4-7	Keyword Hierarchy for Example of Section 4.2.2	80
4-8	Node-to-Key Table for Example	84
4-9	Key-to-Node Table for Example	85
4-10	Forward Ordering Example	89
4-11	Title Word File, Forward Ordering	90
4-12	Reverse Ordering Example	93
4-13	Order Modification Program (Forward Order)	95

Figure	Title	Page
4-14	Final (Modified) Orderings	96
4-15	Ordered File Classification Algorithm	99
4-16	Forward and Reverse Classifications of Sample File	102
5-1	Keys per Cell, Small Keyword File	111
5-2	Keys per Cell, Large Keyword File	112
5-3	Keys per Cell, Title Word File	113
5-4	Cells Searched, Small Keyword File	121
5-5	Cells Searched, Large Keyword File	123
5-6	Cells Searched, Title Word File	124
5-7	Documents Searched, Small Keyword File	127
5-8	Documents Searched, Large Keyword File	128
5-9	Documents Searched, Title Word File	129
5-10	Size of Key-to-Node Table, Small Keyword File	134
5-11	Size of Key-to-Node Table, Large Keyword File	135
5-12	Size of Key-to-Node Table, Title Word File	136
5-13	Sample Nodes of Hierarchy, I	139
5-14	Sample Nodes of Hierarchy, II	140
5-15	Sample Nodes of Hierarchy, III	141
5-16	Portion of Hierarchy Tree, I	143
5-17	Portion of Hierarchy Tree, II	144
5-18	Portion of Terminal Node 1.1.1.1.1.1	146
A-1	Macro-flowchart of Keyword File Preparation	161
A-2	200 Most Frequently Occurring Keywords -	

Figure	Title	Page
	46821 Documents	162
A-3	Alphabetic Listing of Keywords - 46821 Documents	164
A-4	Suffix Deletion Routine	170
A-5	200 Most Frequently Occurring Title Word Stems - 46942 Documents	173
A-6	Rank-Frequency Distribution of Keywords	175
A-7	Log-Probability Plot of Keyword Distribution	176
A-8	Distribution of Number of Keywords per Document	178
B-1	Typical Retrieval Requests	181
B-2	Retrieved Documents Distribution	185

LIST OF TABLES

Table	Title	Page
1-1	Illustration of Typical IS&R Systems	13
1-2	Illustration of Magnitude of Directory and Retrieval with Inverted File Methodology	13
2-1	Definitions of Parameters	16
2-2	Illustration of Magnitude of Directory and Retrieval with Automatic Classification	23
2-3	Characteristics of Typical Mass Storage Devices	32, 33
3-1a	Automatic Classification Experiments	46
3-1b	Automatic Classification Experiments	47
3-1c	Automatic Classification Experiments	48
3-1d	Automatic Classification Experiments	49
3-2	Prior Experiments using CLASFY	52
3-3	Summary of Present Experiments	53
4-1	Ordering Modification Statistics	97
5-1	File Statistics	106
5-2	Percentage of Documents and Cells Searched, CLASFY	131
6-1	Classification Time for 10^6 to 10^7 Documents using CLASFY	154
A-1	Steps Involved in Processing Title Word File	167
A-2	Common Word Comparison	168
A-3	File Statistics	174
B-1	Request Statistics	182

Table	Title	Page
B-2	Documents Retrieved	184

BIBLIOGRAPHY

1. Adams, W. M. and Lockley, L. C., "The preference of seismologists for the KWIC index," prepared for the National Science Foundatio.., (July, 1965).
2. Altmann, B., "A natural language storage and retrieval (ABC) method: its rationale, operation, and further development program," J. Chem. Documentation, 6 (3): 154-157 (Aug. 1966).
3. Angell, T., "Automatic classification as a storage strategy for an information storage and retrieval system," Master's Thesis, The Moore School of Electrical Engineering, Univerity of Pennsylvania (1966).
4. Arnovick, G. N., Liles, J. A., and Wood. J. S., "Information storage and retrieval - analysis of the state of the art," Proc. of the SJCC (1964): 537-61.
5. Atherton, P., "Ranganathan's classification ideas: an analytico-synthetic discussion," Library Resources and Technical Services, 9 (4): 463-73 (Fall 1965).
6. Atherton, P. and Borko, H., "A test of the factor-analytically derived automated classification method applied to descriptions of work and search requests of nuclear physicists," American Institute of Physics Documentation Research Project, Report No. AIP/DRP 65-1; SDC/SP 1905: 1-15 (Jan. 1965).
7. [illegible] B., "Information retrieval based upon

latent class analysis," JACM, 9: 512-21 (Oct. 1962).

8. Baker, F. B., "Latent class analysis as an association model for information retrieval," Symposium on Statistical Methods for Mechanized Documentation (1964): 149-55.

9. Baxendale, P., "Content analysis, specification and control," Annual Review of Information Science and Technology, Vol. I: 71-106 (1966).

10. Berul, L., "Methodology and results of the DOD user needs survey," presented to Special Libraries Assoc. Workshop on the Report Literature (Nov. 2, 1965): 1-24.

11. Berul, L., "Survey of IS&R Equipment," Datamation (March 1968): 27-32.

12. Binford, R. L., "A comparison of keyword-in-context (KWIC) indexing to manual indexing," AD 620 420 (1965).

13. Bonn, T. H., "Mass storage: a broad review," Proc. IEEE, 54 (12): 1861-70 (Dec. 1966).

14. Bonner, R. E., "On some clustering techniques," IBM Journal, 8: 22-32 (Jan. 1964).

15. Borko, H., "The construction of an empirically based mathematically derived classification system," Proc. of the SJCC (1962): 279-89.

16. Borko, H., "Research in automatic generation of classification systems," Proc. of the SJCC (1964): 529-35.

17. Borko, H., "Measuring the reliability of subject classification by men and machines," Amer. Documentation (Oct. 1964): 268-73.

18. Borko, H., "The conceptual foundations of information systems," Systems Development Corp., Report No. SP-2057: 1-37 (May 6, 1965).

19. Borko, H., "Indexing and classification," in H. Borko, Automated Language Processing (John Wiley and Sons, Inc., New York, 1967), pp. 99-125.

20. Borko, H. and Bernick, M., "Automatic document classification," JACM, 10: 151-62 (April 1963).

21. Borko, H. and Bernick, M., "Automatic document classification. Part II. Additional experiments," JACM, 11: 138-51 (April 1964).

22. Bourne, C. P., "Evaluation of indexing systems," Annual Review of Information Science and Technology, Vol. I: 171-90 (1966).

23. Buchholz, W., "File organization and addressing," IBM Systems Journal, 2: 86-111 (June 1963).

24. Buscher, W. C., "The analysis of medical documents with a comparative evaluation of the indexing procedures," PB 169 698 (July 1963), pp. 1-13.

25. Bush, V., "As we may think," Atl. Monthly, 176: 101-8 (July 1945).

26. Bush, V., "Memex revisited," in Science is not Enough (New York, 1967), pp. 75-101.

27. Cane, M., "The dictionary lookup system," in Information Storage and Retrieval, Harvard Computation Laboratory Report No. ISR-9, Section V (Aug. 1965).

28. Chien, R. T. and Preparata, F. P., "Search strategy and file organization in computerized information retrieval systems with mass memory," Coordinated Sci. Laboratory (1967): 1-11.

29. Chonez, N., "Permuted title or key-phrase indexes and the limiting of documentalist work needs," Inform. Stor. Retr., 4 (2): 161-6 (June 1968).

30. Cleverdon, C., Mills, J., and Keen, M., "Factors determining the performance of indexing systems," in ASLIB Cranfield Research Project (Cranfield, Eng., 1966).

31. Computer Command and Control Company, "An automatic classification system to aid R&D management," Report No. 26-104-5 to the Office of Naval Research, Contract NOnr 4531(00) (Nov. 1, 1965).

32. Costello, J. C., Jr., "Storage and retrieval of chemical research and patent information by links and roles in Du Pont," Amer. Documentation, 12: 111-20 (April 1961).

33. Craver, J. S., "A review of electromechanical mass storage," Datamation (July 1966): 22-28.

34. Cuadra, C. A. and Katter, R. V., "Opening the black box of 'relevance,'" J. Documentation, 23: 291-303 (1967).

35. Cuadra, C. A. and Katter, R. V., "The relevance of relevance assessment," Proc. Amer. Documentation Inst., Vol. 4: 95-99 (1967).

36. Curtice, R. M., "Magnetic tape and disc file organizations for retrieval," PB 173 218 (May 1966), pp. 1-44.

37. Dale, A. G. and Dale, N., "Some clumping experiments for associative document retrieval," Amer. Documentation (Jan. 1965): 5-9.

38. Dale, N., "Automatic classification system user's manual," Linguistics Research Center (Univ. of Texas), Report No. LRC 64 TTM-1 (Nov. 1964).

39. Dattola, R. T., "A fast algorithm for automatic classification," in Information Storage and Retrieval, Cornell Univ., Dept. of Comp. Sci. Report No. ISR-14, Section V (Oct. 1968).

40. Debons, A., Scheffler, F. L., and Snide, J. D., "Development and experimental evaluation of a retrieval system for Air Force control-display information," AD 663 756 (Nov. 1967), . .68.

41. Dewey Decimal Classification and Relative Index, 17th edition (Forrest Press, Inc., New York, 1965).

42. Doyle, L. B., "Some compromises between word grouping and document grouping," Symp. on Statistical Association Methods for Mechanized Documentation (1964): 15-24.

43. Doyle, L. B., "Is automatic classification a reasonable application of statistical analysis of text?," JACM, 12: 473-89 (Oct. 1965).

44. Doyle, L. B., "Breaking the cost barrier in automatic classification," AD 636 837 (July 1966).

45. Edwards, J. S., "Adaptive man-machine interaction in information retrieval," Doctoral Dissertation, University of Pennsylvania (1967).

46. European Atomic Energy Community, "Euratom-Thesaurus. Part I. Indexing terms used within Euratom's nuclear documentation system," 2nd ed., Report No. EUR 500.e (1966).

47. European Atomic Energy Community, "Euratom-Thesaurus. Part II. Terminology charts used in Euratom's nuclear documentation system," 2nd ed., Report No. EUR 500.e (1967).

48. Fossum, E. G. and Kaskey, G., "Optimization and standardization of information retrieval language and systems," AD 630 797 (Jan. 1966), pp. 1-87.

49. Freeman, R. R., "Computers and classification systems," J. Documentation, 20 (3): 137-45 (Sept. 1965).

50. Freeman, R. R., "Research project for the evaluation of the UDC as the indexing language for a mechanized reference retrieval system: an introduction," Amer. Inst. Physics Documentation Research Project, Report No. AIP/DRP UDC-1: 4-5 (Oct. 1, 1965).

51. Freeman, R. R., "Research project for the evaluation of the UDC as the indexing language for a mechaniz i reference retrieval system: progress report for the period July 1, 1965 - Jan. 31, 1966," Amer. Inst. Physics Documentation Research Project, Report No. AIP/DRP UDC-2: 1-14 (Feb. 1, 1966).

52. Freeman, R. R., "Evaluation of the retrieval of metallurgical document references using the universal decimal classification in a computer-based system," Amer. Inst. Physics, UDC Project, Report No. AIP/UDC-6: 1-12, 33-50 (April 1968).

53. Freeman, R. R. and Atherton, P., "File organization and search strategy using the universal decimal classification in mechanized reference retrieval systems," Amer. Inst. Physics, UDC Project, Report No. AIP/UDC-5: 1-30 (Sept. 15, 1967).

54. Freeman R. R. and Atherton, P., "Audacious - an experiment with an on-line, interactive reference retrieval system using the universal decimal classification as the index language in the field of nuclear science," Amer. Inst. Physics, UDC Project, Report No. AIP/UDC-7: 1-11, 25-34 (April 1968).

55. Guiliano, V. E. and Jones, P. E., "Study and test of a methodology for laboratory evaluation of message retrieval systems," AD 642 829 (Aug. 1966), pp. 1-183.

56. Gotlieb, C. C. and Kumar, S., "Semantic clustering of

index terms," JACM, 15 (4): 493-513 (Oct. 1968).

57. Grauer, R. T. and Messier, M., "An evaluation of Rocchio's clustering algorithm, in Information Storage and Retrieval, Cornell Univ., Dept. of Comp. Sci. Report No. ISR-12, Section VI (June 1967).

58. Heald, J. H., "The making of test thesaurus of engineering and scientific terms," AD 661 001 (Nov. 1967).

59. Henderson, M. M., "Evaluation of information systems: a selected bibliography with informative abstracts," National Bureau of Standards, Technical Note 297 (Dec. 1967).

60. Houston, N. and Wall, E., "The distribution of term usage in manipulative indexes," Amer. Documentation (April 1964): 105-14.

61. International Business Machines, "Random number generation and testing," Data Processing Techniques (C20-8011): 1-12 (1959).

62. International Business Machines, "Index organization for information retrieval," Data Processing Techniques (C20-8062): 1-63 (1961).

63. International Business Machines, "Introduction to IBM System/360 direct access storage devices and organization methods," Form C20-1649-1: 1-70 (1966).

64. International Business Machines, "Sorting Techniques," Data Processing Techniques (C20-1639-0): 1-99 (1966).

65. International Business Machines, "IBM System/360 operating system, Sort/Merge," Form C28-6543-4: 1-77 (1967).

66. Janning, E. A., "Operations of a document retrieval system using a controlled vocabulary," AD 633 614 (March 1966), pp. 1-20.

67. Kershaw, G. and Davis, J. E., "Mechanization in defense libraries," Datamation (Jan. 1968): 48-53.

68. Knable, J. P., "An experiment comparing key words found in indexes and abstracts prepared by humans with those in titles," Amer. Documentation 16 (2): 123-4 (April 1965).

69. Kraft, D. H., "A comparison of keyword-in-context (KWIC) indexing of titles with a subject heading classification system," Amer. Documentation (Jan. 1964): 48-52.

70. Kucera, H. and Francis, W. N., Computational Analysis of Present-Day American English (Brown Univ. Press, Providence, Rhode Island, 1967).

71. Lancaster, F. W., "Evaluating the small information retrieval system," J. Chem. Documentation, 5 (3): 150-60 (Aug. 1966).

72. Lance, G. N. and Williams, W. T., "A general theory of classificatory sorting strategies. 1. Hierarchical systems," Computer Journal, 9 (4): 373-80 (Feb. 1967).

73. Lance, G. N. and Williams, W. T., "A general theory of classificatory sorting strategies. II. Clustering systems," Computer Journal, 10 (3): 271-7 (Nov. 1967).

74. Lefkovitz, D., "The application of the digital computer to the problem of a document classification system," in Colloquiem on Technical Preconditions for Retrieval Center Operations (April 1964): 133-46.

75. Lefkovitz, D., "Automatic stratification of descriptors," Doctoral Dissertation, University of Pennsylvania (1964).

76. Lefkovitz, D., File Structures for On-Line Systems (Spartan Books, Mar. 1969).

77. Lefkovitz, D. and Angell, T., "Experiments in automatic classification," Computer Command and Control Company Report No. 85-104-6 to the Office of Naval Research, Contract NOnr 4531(00) (Dec. 31, 1966).

78. Lefkovitz, D. and Powers, R. V., "A list-structured chemical information retrieval system," Proc, 3rd National Colloquiem on Information Retrieval (1966): 109-29.

79. Lefkovitz, D. and Prywes, N. S., "Automatic stratification of information," Proc. of the SJCC (1963): 229-40.

80. Lefkovitz, D. and Van Meter, C. T., "An experimental real time chemical information system," J. Chem.

Documentation, 6: 173-83 (Aug. 1966).

81. Lesk, M. E., "Operating instructions for the SMART text processing and document retrieval system," in Information Storage and Retrieval, Cornell Univ., Dept. of Comp. Sci. Report No. ISR-11, Section II (June 1966).

82. Lesk, M. E., "Performance of automatic information systems," Inform. Stor. Retr., 4 (2): 201-18 (June 1968).

83. Litofsky, F. S., Literature Chemist, Private Communication, 1968.

84. Maron, M. E., "Automatic indexing: an experimental inquiry," JACM, 8 (3): 404-17 (July 1961).

85. Mc Eowen, J. R., "PAL - a design for a personal automated library," Doctoral Dissertation, University of Pennsylvania (1969).

86. Montague, B. A., "Patent indexing by concept coordination using links and roles," Amer. Documentation, 13: 104-11 (Jan. 1962).

87. Nagy, G., "State of the art in pattern recognition," Proc. IEEE, 56 (5): 836-62 (May 1968).

88. Needham, R. M., "Application of the theory of clumps," Mechanical Translation and Computational Linguistics, 8: 113-27 (1965).

89. Needham, R. M., "Automatic classification in linguistics," AD 644 961 (Dec. 1966).

90. Needham, R. M. and Sparck Jones, K., "Keywords and clumps," J. Documentation, 20: 5-15 (March 1964).

91. O'Connor, J., "The possibilities of document grouping for reducing retrieval storage size and search time," Advances in Documentation and Library Science, Vol. III: 237-79 (1960).

92. Office of Naval Research, "DOD manual for building a technical thesaurus," AD 633 279 (April 1966), pp. 1-21.

93. Perriens, M. P. and Williams, J. H., Jr., "Computer classification of intelligence-type documents," AD 820 801 (Sept. 1967), pp. 1-92

94. Price, D. J. de S., Science since Babylon (New Haven, Yale University Press, 1961), page 97.

95. Price, D. J. de S., "Nations can publish or perish," Science and Technology (Oct. 1967): 84-99.

96. Price, N. and Schiminovich, S., "A clustering experiment: first step towards a computer-generated classification scheme," Inform Stor. Retr., 4: 271-80 (1968).

97. Prywes, N. S., "Browsing in an automated library through remote access," in Computer Augmentation of Human Reasoning (June 1964), pp. 105-30.

98. Prywes, N. S., "An information center for effective R&D management," Proc. 2nd Congr. on Information Systems Science (Nov. 1964): 109-16.

99. Prywes, N. S., "Man-computer problem solving with

multilist," Proc. IEEE, 54: 1788-1801 (Dec. 1966).

100. Prywes, N. S., "Information storage and retrieval: extending human memory and recall," in J. Rose, ed., Survey of Cybernetics (July 1968).

101. Prywes, N. S., "On-line information storage and retrieval," AGARD Symp. on Storage and Retrieval of Information (June 18-21, 1968): 1-18.

102. Prywes, N. S., "Structure and organization of very large data bases," Proc. Symp. on Critical Factors in Data Management, UCLA (March 1968).

103. Prywes, N. S. and Gray, H. J., "The organization of a multilist-type associative memory," Gigacycle Computing Systems, AIEE General Meeting (Jan. 1962): 87-101.

104. Prywes, N. S., et al., "The multi-list system technical report No. 1," AD 270 573 (Nov. 1961), pp. 1-100.

105. Prywes, N. S., et al., "The multi-list system technical report No. 1 (Part II and III)," AD 270 572 (Nov. 1961) pp. 101-238.

106. Radio Corporation of America, "Spectra 70, random access devices," Form 70-06-500 (Nov. 1967).

107. Ranganathan, S. R., "The colon classification." in S. Artandi, ed., Rutgers Series on Systems for the Intellectual Organization of Information, Vol IV (Rutgers Univ. Press, New Brunswick, New Jersey, 1965), pp. 9-298.

108. Rees, A. M., "Evaluation of information systems and services," Annual Review of Information Science and Technology, Vol II: 63-86 (1967).

109. Resnick, A., "Relative effectiveness of document titles and abstracts for determining relevance of documents," Science, 134: 1004-6 (1961).

110. Rocchio, J. J., Jr., "Document retrieval systems optimization and evaluation," Doctoral Dissertation, Division of Engineering and Applied Physics, Harvard University (1966).

111. Rolling, L. N., "A computer-aided information service for nuclear science and technology," J. Documentation, 22 (2): 93-115 (1966).

112. Ross, I. C., "BE PIPER, the 1964 model of the Bell Laboratories' permutation index program for the IBM 7090/94," Unpublished Technical Memo, MM 64-1222-8 (Sept. 1964), pp. 1-9.

113. Rubinoff, M., Bergman, S., Franks, W., and Rubinoff, E. R., "Experimental evaluation of information retrieval through a teletypewriter," AD 660 083 (June 1967), pp. 1-25.

114. Rubinoff, M. and Stone, D. C., "Semantic tools in information retrieval," AD 660 087 (May 1967), pp. 1-17.

115. Russell, M. and Freeman, R. R., "Computer-aided indexing of a scientific abstracts journal by the UDC with UNIDEK: a case study," Amer. Inst. Physics, UDC

Project, Report No. AIP/UDC-4: 1-20 (April 1, 1967).

116. Salton, G., "Progress in automatic information retrieval," IEEE Spectrum, 2: 90-103 (Aug. 1965).

117. Salton, G., ed., Information Storage and Retrieval, Cornell Univ. Dept. of Comp. Sci., Report No. ISR-12 (Aug 1967).

118. Salton, G., ed., Information Storage and Retrieval, Cornell Univ. Dept. of Comp. Sci., Report No. ISR-13 (Dec. 1967).

119. Salton, G. and Lesk, M. E., "Information analysis and dictionary construction," in Information Storage and Retrieval, Cornell Univ. Dept. of Comp. Sci., Report No. ISR-11, Section IV (June 1966).

120. Salton, G and Lesk, M. E., "Computer evaluation of indexing and text processing," JACM, 15 (1): 8-36 (1968).

121. Saracevic, T., "Quo vadis test and evaluation," Proc. Amer. Documentation Inst., Vol. IV: 100-04 (1967).

122. Scheffler, F. L., "Indexer performance analysis and operations of a document retrieval system," University of Dayton Research Institute (Dayton, Ohio), Technical Report AFML-TR-67-379: 1-76 (Feb. 1968).

123. Sinnett, J. D., "An evaluation of links and roles used in information retrieval," AD 432 198 (Dec. 1963), pp. 1-300.

124. Soergel. D., "Mathematical analysis of documentation

systems. An attempt to a theory of classification and search request formulation," Inform. Stor. Retr., 3: 129-73 (July 1967).

125. Sokal, R. R., "Numerical taxonomy," Sci. Am., 215: 106-16 (Dec. 1966).

126. Sparck Jones, K. and Jackson, D., "Current approaches to classification and clump-finding at the Cambridge Language Research Unit," Computer J. (May 1967): 29-37.

127. Sparck Jones, K. and Needham, R. M., "Automatic term classifications and retrieval," Inform. Stor. Retr., 4 (2): 91-100 (June 1968).

128. Stevens, M. E., "Automatic indexing: a state-of-the-art report," NBS Monograph No. 91 (March 1967), pp. 1-220.

129. Stone, D. C., "Word association experiments - basic consideration," AD 660 085 (Aug. 1966), pp. 2-15.

130. Stone, D. C., "Word statistics in the generation of semantic tools for information systems," AD 664 915 (Dec. 1967).

131. Stone, P. J., Dunphy, D. C., Smith, M. S., and Ogilvie, D. M., The General Inquirer: A Computer Approach to Content Analysis (M. I. T. Press, Cambridge, Mass.)

132. Stone, P. J., et al., User's Manual for the General Inquirer (M. I. T. Press, Cambridge, Mass.), pp. 25-9.

133. Swanson, D. R., "The evidence underlying the Cranfield results," Library Quarterly, 35: 1-20 (Jan. 1965).

134. Taube, M., "Notes on the use of roles and links in coordinate indexing," Amer. Documentation (Apr. 1961): 98-100.

135. Taube, M., "A note on the pseudo-mathematics of relevance," Amer. Documentation 16 (2): 69-72 (Apr. 1965).

136. Taulbee, O. E., "Classification in information storage and retrieval," Proc. 20th ACM Conf. (Aug. 1965): 119-37.

137. Thesaurus of Engineering and Scientific Terms, 1st ed., (Engineers Joint Council, New York, Dec. 1967).

138. Thompson, D. A., Bennigson, L., and Whitman, D., "Structuring information bases to minimize user search time," Proc. Amer. Documentation Inst., Vol IV: 164-8 (1967).

139. Thompson, D. A., Bennigson, L., and Whitman, D., "A proposed structure for displayed information to minimize search time through a data base," Amer. Documentation (Jan. 1968): 80-84.

140. United States Atomic Energy Commission, "Descriptive cataloging guide," Division of Technical Information Report No. TID-4577 (Rev. 2) (July 1966).

141. United States Atomic Energy Commission, "Subject headings used by the USAEC Division of Technical Information," Division of Technical Information Report No. TID-6001 (6th rev.) (Nov. 1966).

142. United States Atomic Energy Commission, "Magnetic tape

formats for AEC entry and keyword files," Division of Technical Information Ext., Computer Operations Branch (rev. 1) (Oct. 1967).

143. Van Meter, C. T., Bedrosian, S. D., and Lefkovitz, D., et al., "CIDS No. 3, a proposed chemical information and data system," Inst. for Cooperative Research, University of Pennsylvania (Dec. 1965).

144. Van Oot, J. G., Schultz, J. L., McFarlane, R. E., Kvalnes, F. H., and Riester, A. W., "Links and roles in coordinate indexing and searching: an economic study of their use, and an evaluation of their effect on relevance and recall," J. Chem. Documentation, 6 (2): 95-101 (May 1966).

145. Vickery, B. C., On Retrieval System Theory, 2nd ed. (Butterworth and Co., London, 1965).

146. Vickery, B. C., "Faceted classification schemes," in Rutgers Series on Systems for the Intellectual Organization of Information, Vol. V, pp. 9-108 (1966).

147. Walston, C. E., "Information retrieval," in Advances in Computers, Vol. VI, pp. 1-30 (1965).

148. Ward, J. H., Jr. and Hook, M. E., "Application of a hierarchical grouping procedure to a problem of grouping profiles," Education and Psychological Measurement, 23: 69-92 (1963).

149. Weintraub, S., Tables of the Cumulative Binomial Probability Distribution for Small Values of P (The Free

Press of Gencoe, London, 1963).

150. Williams, J. H. Jr., "A discriminant method for automatically classifying documents," Proc. of the FJCC: 161-6 (1963).

151. Williams, J. H., Jr., "Annual progress report - computer classification of documents," AD 663 178 (June 1967), pp. 1-19.

152. Williams, "Fundamentals of indexing," in Principles of Automated Information Retrieval, Chapter 5, pp. 115-44.

153. Wolfberg, M. S., "Determination of maximally complete subgraphs," The Moore School of Electrical Engineering, University of Pennsylvania, Report No. 65-27 (May 1965).

154. Zimmerman, B., "Automatic classification for the ASTIA mathematics collection," Master's Thesis, The Moore School of Electrical Engineering, University of Pennsylvania (1964).

155. Zipf, G. K., Human Behavior and the Principles of Least Effort (Addison-Wesley Press, Inc. Cambridge, Mass., 1949).

CHAPTER 1

INTRODUCTION TO INFORMATION STORAGE AND RETRIEVAL

1.1 Magnitude of the Problem

The "information explosion" is here to stay and anyone designing an information storage and retrieval system must take cognizance of that fact. Not only is the current publication rate high - about 350,000 scientific papers per year [95] - but the growth in this rate is staggering. De Solla Price [94] has plotted the number of scientific journals (see Fig. 1-1) published each year from the oldest surviving journal, Philosophical Transactions of the Royal Society of London (1665), until today, when we are rapidly approaching 100,000 journals.

In 1830, in order for scientists to keep up with the increasing number of papers in the 300 journals of that day, the first abstract journal was introduced. Since then, as can be seen in Fig. 1-1, the growth of abstract journals has essentially matched that of primary journals. We are now long past the point of 300 abstract journals being published yearly without a solution comparable to the one found in 1830.

Thus, any solution to the problem of collecting information and supplying desired information in the proper amounts to the proper people at the proper time must be able to handle this rapid growth in literature

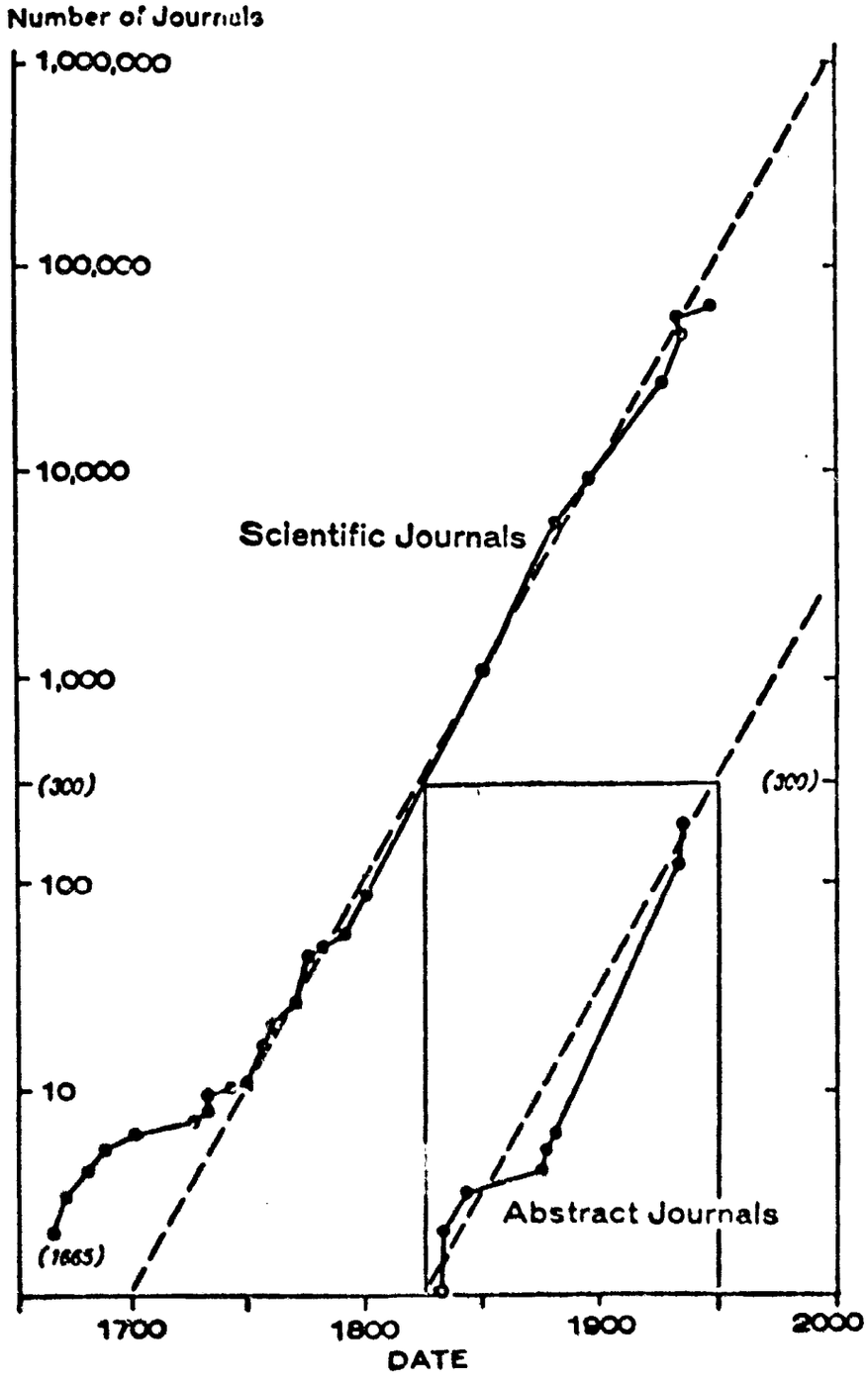


Figure 1-1

Growth of Journals and Abstract Journals
(from de Solla Price [94])

as well as th substantial body of publications which have been and which are being produced in each of the technical fields today.

1.2 IS&R Functions

The objective of all IS&R systems should be to serve a given community by supplying desired information upon request. By the word "serve" is meant that the system should function at the convenience of the user. This implies simplicity of use, multiple modes of use to suit each type of request, and accurate and prompt responses. The degree to which a system meets this objective is determined by how it is set up, how it is used, and how much money is available for system design and operation.

The first parameter that must be considered in a system of this type is the size of the collection. While there are a number of instances where small collections might be useful (i.e., a library of a small company in a very specialized area or a personal information system [25,26,85]), considering the previous section it can be seen that there must be a substantial number of systems which, presently or in the near future, are or will be concerned with large collections of information. It is to these systems that this dissertation is directed. Henceforth, all references to IS&R systems will automatically imply systems with information collec-

tions of, at the very least, tens of thousands of items.

IS&R systems can handle a variety of types of information, ranging from individual scientific facts to merchandise in an inventory to journal articles and books in a library. For convenience and simplicity the discussions and examples following will be restricted to references to libraries with the understanding that "information items" or "documents" refer to any of the publications usually found in libraries.

1.2.1 Storage

The storage functions of an IS&R system are shown in Fig. 1-2. The acquisition of documents for a collection can be more than just deciding as to the pertinence of a particular document to the collection. If documents are acquired in the proper formats, later stages in the storage process could be easily automated. Advantage should be taken of advances in computerized typesetting and optical character readers [11] to facilitate automatic processing of documents.

The purpose of the indexing function is to obtain a number of descriptors which act as a surrogate for the document. These descriptors, or keywords, can be obtained manually or automatically by computer analysis of the document title, abstract or text. The indexing function will be discussed in greater detail later in this paper.

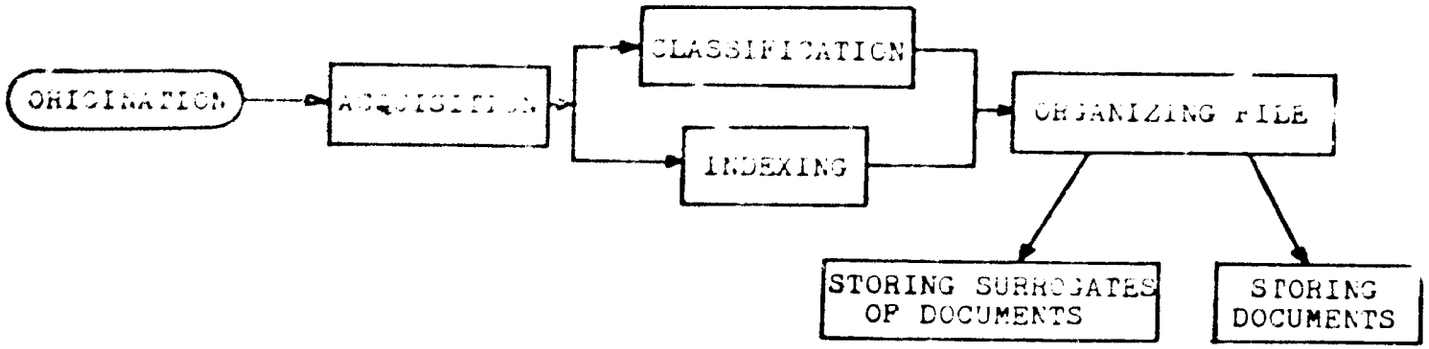


Figure 1-2

Typical Storage of Documents

Classification of documents refers to the grouping of like documents into categories. The categories can be set up independent of the documents (a priori) or after all the documents have been indexed (a posteriori). In the former case, document classification can be done manually or automatically at the same time as indexing. In the latter case it is done only after all the documents have been indexed and will probably be an automatic process. Most IS&R systems make no or very little use of document classification.

Organizing the document file involves setting up of directories and deciding the order of the documents in the file. Parts of this step are closely related to document classification, particularly in an IS&R system employing a posteriori automatic classification.

Document storage [11] and surrogate storage is done separately because the documents themselves do not have to be accessed as rapidly as their surrogates. Most systems do not involve automatic document storage. However, this should play an increasing role in large scale automated IS&R systems. Surrogate storage can include the above mentioned keyword surrogates, titles, authors, abstracts and/or other items (called association terms by Prywes [100]) deemed to be of use in deciding the applicability of documents to retrieval requests.

1.2.2 Retrieval

The basic retrieval functions of an IS&R system are shown in Fig. 1-3. The user formulates a request and submits it to the system. This request could range from a well thought out query to a vague notion of what is desired. In fact, the user might not know what he is looking for at all; he might just be browsing through the collection to see if he can come up with anything of interest. Also, the user might have need for every item pertaining to his request, as in a patent search, or he might be perfectly satisfied with one or a few such items.

In order to satisfy all of the above needs an IS&R system must be able to provide the user with the option to refine his request after various stages of preliminary processing [71]. According to a literature chemist in a non-automated library [83], the ability to go back to the user in order to refine the request is desirable in all, very helpful in many, and absolutely essential in some literature searches. Because of the rapidly changing natures of thought processes and user needs this request refinement should take place immediately after the original request is submitted. This implies that an effective IS&R system should include on-line man-machine interaction [18,97,113].

Path A of Fig. 1-3 includes data such as pre-

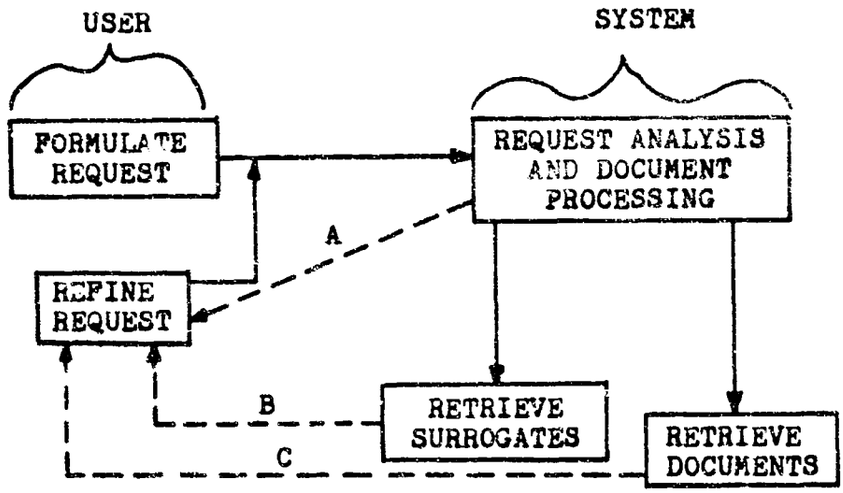


Figure 1-3

Retrieval of Documents

liminary document counts, suggested query modifications, classification hierarchy display, and other data which might be available before accessing either the surrogate or the document storages. In most of today's systems this very important function is limited to (non-automated) discussions with a professional librarian or to automated document counts or more often, it does not appear at all.

Path B essentially exists only in on-line systems. This assumes that request refinement done after a day or two's wait is almost identical with new request submission. The advantages of Path B can be enhanced if upon command surrogates of documents similar to those requested can be displayed. This is relatively easy to do in a classified collection.

Path C can only exist if the documents are stored and can be retrieved or have parts of them displayed rapidly. This feature is rare today but should be incorporated in future systems.

1.3 Inadequacies of Current Systems

A significant portion of the library problem is that most of today's "automated" IS&R systems are hardly automated at all. On the storage side, indexing and cataloging are the main areas which should be switched from the human to the computer domain. At a recent symposium on IS&R, Prywes [101] stated:

> "In any one of the large libraries or information centers there are thousands of monographs and serials that are waiting to be catalogued and indexed. These often lay unused because of the dearth of competent cataloguers and indexers, especially those expert in particular subjects and languages. The increased amount of material which is being circulated soon may require substantial increase in staff. Staff with this competence is extremely scarce; low salaries discourage young people from library work. For these reasons the storage process tends to constitute a serious bottleneck."

One is tempted to draw an analogy between automation of libraries today and automation of the telephone network some years ago. It is said that if the dial did not replace telephone operators, all the women in this country would have difficulty in handling today's volume of telephone traffic.

Computer processing of natural language text for indexing, and automatic classification for cataloging can break this bottleneck.

On the retrieval side, libraries and information centers operate at low levels of effectiveness and are called upon as information sources a relatively low percentage of the time information is required [10]. One reason for this is the indirect route one must use to use IS&R systems. On-line interactive systems could solve much of this problem.

There are also problems with the automated portion of IS&R systems. Present systems are generally ineffi-

cient and are restricted to few modes of operation (i.e., none allow a reasonable degree of mechanized browsing). Most systems in use or being proposed today are either of the serial, inverted file or list-organized types. Each of these systems has advantages and should be used in certain situations. However, each has serious drawbacks when used for retrieval by combination of document descriptors (keywords).

The main difficulty with a serial file is the time required to access information. Even with the high speed computers of the foreseeable future, search times through serial files of millions of documents will be on the order of many minutes or even hours. This leads to the need for batching many requests and eliminates the possibility of on-line, real-time information retrieval.*

In standard inverted file systems, the document surrogates are stored in any order, usually by accession number. Lists are maintained in a directory for accessing this file. There is one list per keyword and the entries in the lists are pointers to document surrogates. Retrievals are performed by logical comparisons between

---

* There are methods to shorten serial searches, though not by enough to upgrade serial systems to the real-time domain. Fossum and Kaskey [48] inquire as to the efficacy of certain of these methods. Their questions are answered near the end of Chapter 5 of this paper.

directory lists as specified by queries. Tables 1-1 and 1-2, modified from Prywes [102], give examples of parameters to be encountered in typical IS&R systems.

As may be seen in item I of Table 1-2, directory size in an inverted file system may range from ten million to close to a billion words. These must be stored on relatively fast and expensive media (i.e., disks rather than magnetic cards or strips) because of the necessity of frequent access. In fact, the number of directory words required per query (item J in Table 1-2) might be so large compared to available high speed storage as to require multiple accessions of the same lists in order to process a query. A method which reduces these quantities by an order of magnitude or more is shown in the next chapter of this paper.

A recent study of mechanization in defense libraries [67] points out some of the shortcomings in current systems. All of the systems studied (27 in all) use either serial files or inverted files. None use automatic indexing or automatic classifications.

List-organized document retrieval systems chain document surrogates together via keyword lists. In other words, a search on a keyword involves jumping from document to document which contain that keyword. Thus, most of the directory is actually stored in the surrogate file itself. Objections to this type of storage for large scale IS&R

		Number		
A.	Item Records in File	10^6	to	10^7
B.	Keywords assigned to an Item (Av.)	10		50
C.	Keywords in Vocabulary	10^4		5×10^4
D.	Average Number of Items assigned to the same Keyword = (AxB)/C	10^3		10^4
E.	Keywords Specified in a Query (Av.)	10		50
F.	Items referenced by Keywords in a Query (Av.) = E x D	10^4		5×10^5

Table 1-1

Illustration of Typical IS&R Systems

Directory:		Number		
G.	Lists in Inverted File Directory; 1 list per keyword = C	10^4	to	5×10^4
H.	Accession Numbers per list in the Directory (Av.) = D	10^3		10^4
I.	Words in Directory = G x H (Assume 1 computer word per Accession Number)	10^7		5×10^8
Retrieval:				
J.	Inverted File Directory Words Brought from Secondary to Primary Storage = F (All accession numbers in records that correspond to query keywords)	10^4		5×10^5

Table 1-2

Illustration of Magnitude of Directory and Retrieval with Inverted File Methodology

systems center about length of the lists and speed of the storage media required for reasonable search times. Discussion and examples of list-organized systems are available [48,76,99,103,104,105].

CHAPTER 2

CONCEPTS OF IS&R BASED ON AUTOMATIC CLASSIFICATION

2.1 Classification Parameters

The initial goal of classifying documents is to group "like" documents together into categories. The documents (or document surrogates) are then placed near each other to facilitate retrieval. In a conventional library the documents are placed on the same or adjoining shelves. In an automated library the document surrogates (plus room for additions) are placed into convenient units of memory such as cylinders on a disk or magnetic strip or card. These units which include categories of information will be called cells (sometimes called "buckets" [23,28,62]).

It is desirable to have a quantitative measure of the "likeness" of documents. In a collection of documents indexed with keywords, such a measure is supplied by the number of keywords common to two documents. Extending this notion, a measure of the quality of a classification system is how well the classification algorithm minimizes the average number of different keywords in a cell. Definitions of parameters pertinent to this concept are presented in Table 2-1.

Figure 2-1 shows the bounds on N_{KC}, the average number of keys per cell. It must be greater than or equal

Parameter	Definition
N_v	Vocabulary size, total number of different keywords
N_d	Number of documents in the system
N_c	Number of cells in a given classification
N_{kd}	Average number of keys assigned per documents (i.e., average depth of indexing)
N_{kc}	Average number of different keys per cell (this is the quantity to be minimized)
N_{ck}	Average number of cells a given key is assigned to $= \frac{N_c \times N_{kc}}{N_v}$
N_{dc}	Average number of documents per cell $= N_d/N_c$

Table 2-1

Definitions of Parameters

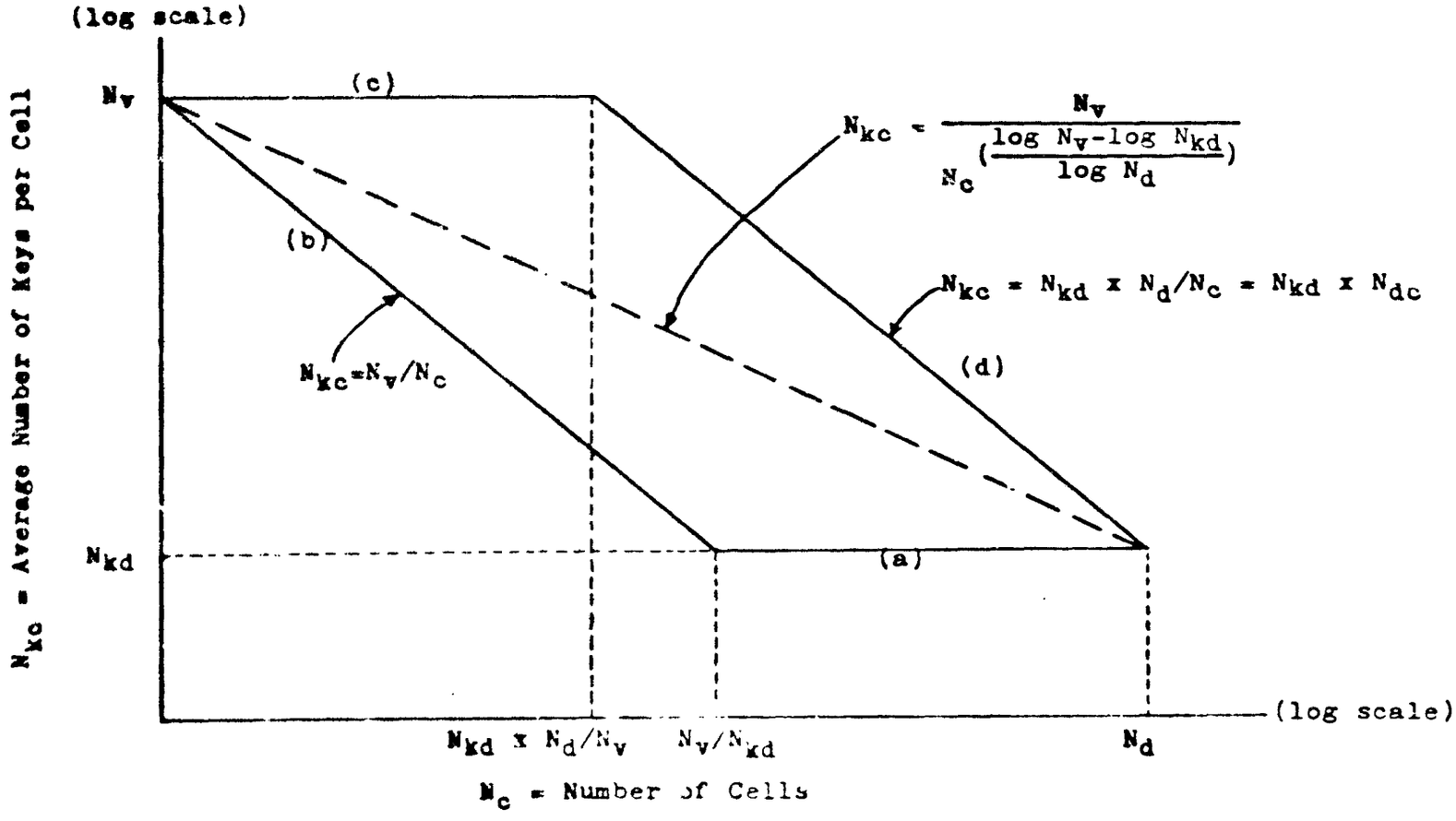


Figure 2-1

Classification Parameters

to the larger of (a) N_{kd}, the average number of keys per document and (b) N_v/N_c, the vocabulary size divided by the number of cells. At the same time, N_{kc} must be less than or equal to the smaller of (c) N_v, the vocabulary size and (d) $N_{kd} \times N_{dc}$, the number of keys per document multiplied by the number of documents per cell. Thus the average number of keys per cell for any given number of cells must fall within the parallelogram of Figure 2-1.

The diagonal dashed line represents the approximate region of the expected plot of N_{kc} vs. N_c for a good classification system. This expectation has been arrived at by past experiments [3,77] and those described in this paper. The actual path of this curve depends not only on the classification algorithm but also upon the collection itself. For example, if all keywords were unique, the curve would follow the upper boundary regardless of the classification algorithm used. Likewise, for any real collection of documents, it would be very unlikely for the curve to even approach the lower boundary.

An interesting point to consider is that Fig. 2-1 shows that serial and inverted files can be considered as special cases of classification. A serial file would have $N_c = 1$ and $N_{kc} = N_v$, thereby occurring at the upper left corner of the diagram. Here, all the documents are in one cell which is searched serially. An inverted file appears at the opposite corner with $N_c = N_d$ and $N_{kc} = N_{kd}$.

Now each cell contains only one document as in an inverted file.

2.2 Advantages of A Posteriori, Hierarchical, Automatic Classification Systems for On-Line Retrieval Systems

Most IS&R systems do not use classification at all. Of those that do, many assign multiple categories to each document and use these categories in place of index terms (example: Universal Decimal Classification, see Freeman [51,52] and Freeman and Atherton [53,54]) or use a unique category assigned to a document as an additional index term. The full potential of classification as an adjunct to indexing has not as yet been approached. The following sections discuss what can be done through the use of the proper type of classification.

2.2.1 Automatic Classification: Directory Size Reduction

The magnitude of the inverted file directory was shown in the previous chapter. The use of automatic classification can reduce the size of this directory by more than an order of magnitude. This is done by forming an inverted file directory on the cells, rather than on the individual documents. It is true that once a cell whose keys satisfy the query has been located, a search must then be made in the cell for applicable documents. However, this is not as much of a hardship as one might think because of the following effects:

1) Cell access time is generally much greater than transmission time. Therefore, it is not very costly to read the contents of an entire cell if one has to access the cell for one or more documents anyhow.

2) Grouping logical document records together into larger physical records can provide significant storage savings.

3) Considerable transmission and processing time (plus, of course, memory costs) are saved by manipulating much shorter directory lists.

4) Memory accesses can be reduced. This is covered in Section 2.2.5 of this paper.

In order to demonstrate the order of magnitude reduction in directory size, consider the sample conditions presented in Section 1.3. The expected N_{kc} vs. N_c curves are shown in Figure 2-2 along with their respective parellelograms. The circles represent numbers of cells chosen for this example. The actual number of cells in an IS&R system will be decided upon by a trade-off between a number of factors including:

1) Cell size should be a convenient multiple or sub-multiple of a suitable storage unit.

2) The fewer the cells the smaller the directory and the fewer the number of directory words which have to be brought into high speed

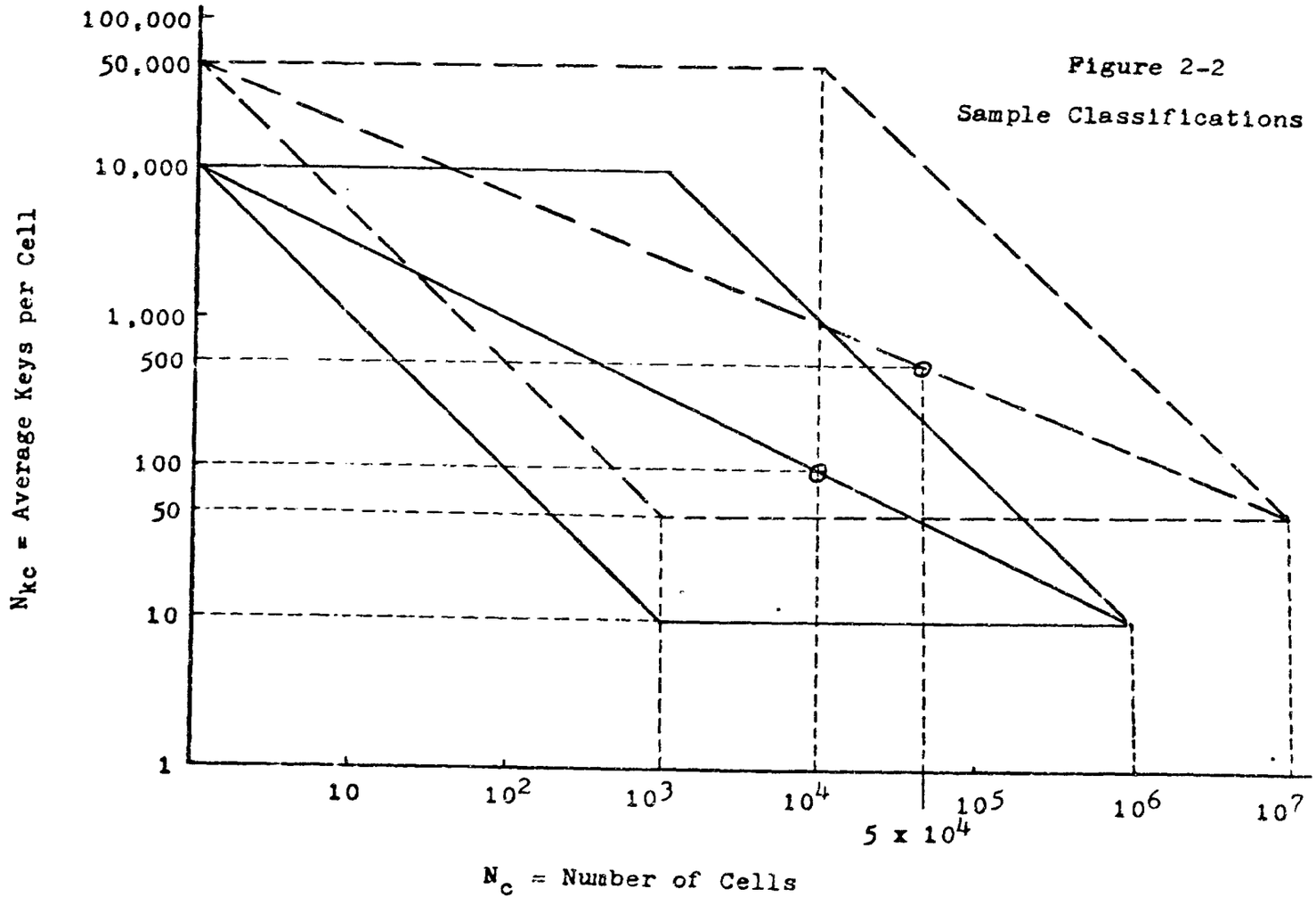


Figure 2-2

Sample Classifications

storage for each request.

3) The more cells there are the fewer number of documents per cell and hence the shorter the search through each cell.

4) More cells mean fewer keys per cell. This increases the selectivity of each cell.

The appropriate numbers for the classified files are given in Table 2-2. This is based on 10,000 and 50,000 cells (N_c) or 100 and 200 documents per cell (N_{dc}) for the two cases respectively. The order of magnitude reduction in directory size (10^7 to 5×10^8 compared with 10^6 to 2.5×10^7) and in words brought from secondary to primary storage (10^4 to 5×10^5 compared with 10^3 to 2.5×10^4) is evident by comparison of Tables 1-2 and 2-2.

2.2.2 Automatic, A Posteriori: Flexible

All of the well-known classification systems of today are a priori systems. In other words, the categories and sub-categories were decided upon on the basis of some "natural" divisions of knowledge and then the documents were (and still are) placed into these categories. Some problems with this traditional point of view are:

1) Few areas of knowledge can be divided in a truly "natural" sense. For example, should biochemistry be a sub-division of biology or of chemistry? The answer to this might depend

Directory:	Number	
N_v = Lists in Automatic Classification Directory (1 list per term)	10^4 to	5×10^4
N_c = Number of cells (Circles in Fig. 2-2)	10^4	5×10^4
N_{kc} = Average number of keywords per cell (Fig. 2-2)	10^2	5×10^2
N_{ck} = Average number of cells per keyword = Average number of words in a directory list (1 computer word per entry)	10^2	5×10^2
Words in directory = $N_v \times N_{ck}$	10^6	2.5×10^7
Retrieval:		
N_{ck} = Cell references per key	10^2	5×10^2
Directory words brought from Secondary to Primary Storage = N_{ck} x keys per query (item E, Table 1-1)	10^3	2.5×10^4

Table 2-2

Illustration of Magnitude of Directory and Retrieval with Automatic Classification

upon the rest of the collection (i.e., whether it is mainly biological or chemical in nature).

2) Overlapping of disciplines is increasing.

3) The classification schedules of a priori systems require significant effort to be kept up to date. The Universal Decimal Classification is an example of one with many outdated structures [54].

4) New areas of knowledge must fit into the existing schedules. This results in highly artificial hierarchies of knowledge. Figure 2-3 shows that in the Dewey decimal classification, electrical engineering is considered a subset of mechanical engineering. It is clear that this came about historically, and not because of today's view of the subdivision of knowledge.

5) Specialized libraries make use of small portions of existing schedules. For instance, the average technical library which uses the Dewey decimal classification probably has 90 percent or more of its documents filed in the 500 (pure science) or 600 (technology) divisions [152]. One effect of this is very deep indexing, such as 621.3841361 for Communication

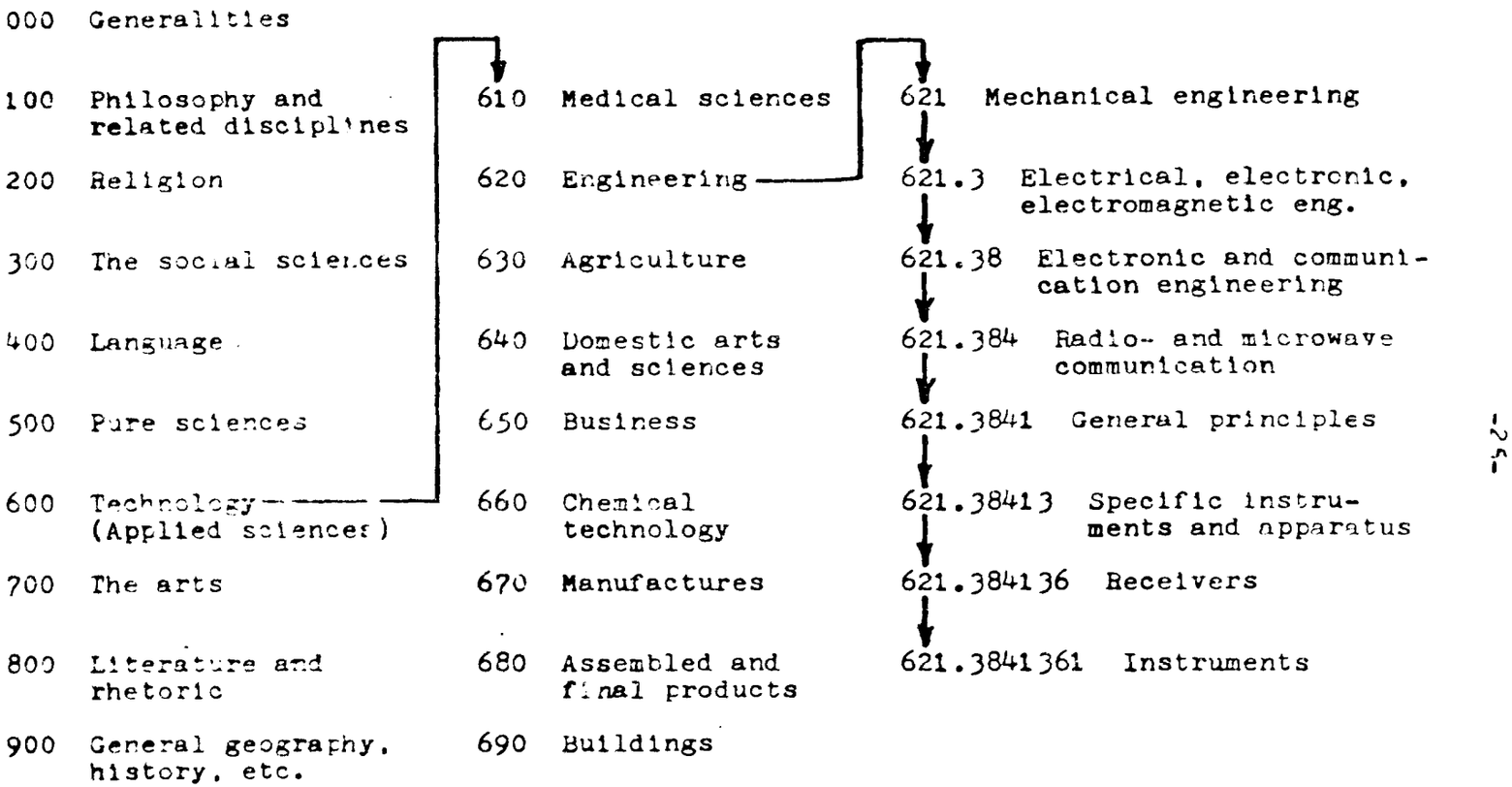


Figure 2-3

Dewey Decimal Classification Schedule [41]

instruments [41].

6) Existing classification trees usually involve at least ten and as many as several thousand alternatives at each decision node. Recent studies have shown that the optimum number of alternatives at each node is usually (depending on certain parameters) considerably less than ten [103,104,138,139].

Our needs for information are changing, therefore our classification schedules must be capable of changing. In an automatic, a posteriori classification system, the categories are decided upon after all the documents have been indexed. In this way, the resulting system is specifically designed for a particular collection at a particular point in time. If there are significant changes in the collection, the system can be automatically reorganized to fully reflect the current status of the collection.

A major objection to automatically derived classification categories is that they might be different from those decided upon by human beings. However, the quality of a system should be measured by its convenience to the user, and not by how the system is originated. Besides, who knows that the human is right, and not the machine?

2.2.3 Hierarchical, On-Line: Browsable

In "The Conceptual Foundations of Information Systems", Borko [18] notes:

> "The user searches for items that are interesting, original, or stimulating. No one can find these for him; he must be able to browse through the data himself. In a library, he wanders among the shelves picking up documents that strike his fancy. An automated information system must provide similar capabilities."

The ability to browse through parts of a collection should be an essential portion of every IS&R system. There are many times when one has only a vague idea of the type of document desired. Browsing can help channel pseudo-random thoughts into a direct line towards the information actually desired.

Effective browsing demands a hierarchical classification system in order to enable one to start with broad categories and work towards specifics. Automatic classification can produce such hierarchical sets of categories. In a priori systems, nodes are given names and index numbers. However, in a posteriori systems the node names are generated automatically and consist of the set of keywords which appear in all the nodes directly beneath (thinking of the hierarchy as an inverted tree) the node in question. This resulting set of keywords can be considered an "abstract" [97] of the knowledge contained beneath that node in the tree. If a set of keywords is too large, humans or preferably automatic

processes can be employed to condense the set and provide a suitable title for the node.

Naturally, automated browsing can only be effective in on line systems through man-machine interaction. The user can enter a node through a conjunction of keywords. The system would then display the nodes beneath the original one as well as some statistics such as how many documents there are beneath each node or how many documents contain each displayed keyword. When the user selects a branch, the cycle repeats with the new node. If desired, one could backtrack up the hierarchy or jump to completely different sections of it. Once the user has narrowed his search, he can demand retrieval of some or all of the documents by specifying keywords and/or categories.

Another way of browsing in a classified set of documents is to start at the very bottom. Assume one has a specific query in mind and upon submitting it to the system, obtains only one document. If this is insufficient one could broaden the search by requesting the display of other documents in the category of the one retrieved. Since these documents are close in content to the original, they might also be satisfactory or their keywords might suggest ways for the user to refine his query in order to retrieve other documents of interest.

None of these modes of browsing could be utilized

by files with strict serial or inverted file organization.

2.2.4 Hierarchical: Further Directory Size Reduction

A small hierarchy is illustrated in Figure 2-4. The keywords are represented by the letters A - H and the nodes 2, 3, and 4 are assumed to be terminal nodes or cells. The directory for the inverted file on cells, called the key-to-cell table, has 15 entries. Figure 2-5 illustrates the hierarchy effect presented in the previous section. Here, the keywords A, B, and C have moved up to node 1 because all one nodes beneath node 1 contained them. At the same time these keywords were deleted from the lower nodes. Now, the key-to-node table has only 9 entries.

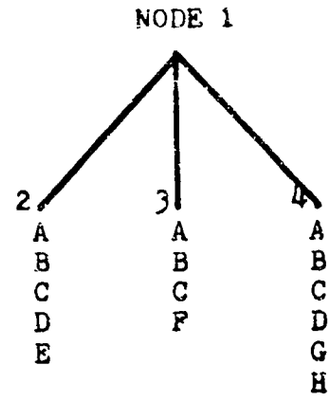
This example illustrates a further reduction in directory size via use of a key-to-node table. Based on experiments to be described later in this paper, this reduction seems to be on the order of about 10-15 percent. This reduction is applied to

(a) the amount of memory required for the directory,

(b) the number of directory words which must be brought into main storage for each query,

and (c) the number of directory words which must be processed for each query.

These benfits are obtained at the cost of increased processing for each directory word. In the example of Figure 2-5, keyword A points to node 1 and keyword G to

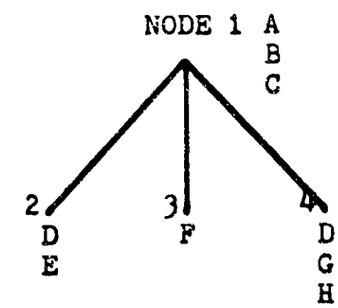


KEY-TO-CELL TABLE

KEYS	A	B	C	D	E	F	G	H
CELLS	2 3 4	2 3 4	2 3 4	2 4	2	3	4	4

Figure 2-4

Inverted File on Cells



KEY-TO-NODE TABLE

KEYS	A	B	C	D	E	F	G	H
CELLS	1	1	1	2 4	2	3	4	4

Figure 2-5

Inverted File on Nodes

node 4. A query involving the conjunction of A and G should indicate node 4. Thus the query decoding program must realize that node 4 is under node 1 in the hierarchy. Proper program design on suitable computers should minimize this task.

2.2.5 On-Line Retrievals: Memory Accesses Reduction

Most mass storage devices have two components to the time required to retrieve a record. The larger component is the time required for the read mechanism to approach the vicinity of the desired information (or vice-versa). This is called the access time and is itself made up of two components, motion access and latency (usually averaging one-half revolution of the recording media). The smaller component of the retrieval time is the actual data transmission time. Typical characteristics of some mass storage devices are shown in Table 2-3. Comparing the total access times with the time required to transmit 2000 bytes, one sees factors ranging from 7 for the smaller devices up to 19 for the larger capacity memories. This illustrates two points:

1) Once the access time has been "spent," it costs relatively little more to read additional data as long as another access time is not involved.

2) An appreciable time savings can be made by

Device	Capacity (millions of bytes)	Data Rate (thousands of bytes/sec.)	Transmission Time-2000 Bytes (milliseconds)	Average Motion Access Time (milliseconds)
IBM 2311 Disk Storage Drive	7.25	156	12.8	75
IBM 2314 Direct Access Storage Facility (8 disk packs)	pack: 29.17 total: 233.40	312	6.4	75
RCA 564 Disk Storage Unit	7.25	156	12.8	75
IBM 2321 Data Cell Drive (tape strips)	400.	55	36.4	350
RCA 568-11 Mass Storage Unit (RACE-magnetic cards)	536.9	70	28.6	508

Table 2-3
(to be continued)

Characteristics of Typical Mass Storage Devices [13,33,63,106]

Device	Average Latency = ½ revolution (milliseconds)	Total Access Time Random Records (milliseconds)	Approximate Cost / Byte (cents)
IBM 2311 Disk Storage Drive	12.5	87.5	.35
IBM 2314 Direct Access Storage Facility (8 disk pack)	12.5	87.5	.10
RCA 564 Disk Storage Unit	12.5	87.5	.35
IBM 2321 Data Cell Drive (tape strips)	25*	375	.033
RCA 568-11 Mass Storage Unit (RACE-magnetic cards)	30*	538	.026

*If an entir. track is read, these times may be omitted.

Table 2-3
(continued)

Characteristics of Typical Mass Storage Devices [13,33,63,106]

reducing the required number of memory accesses.

These points are very pertinent to on-line systems because the lack of the ability to batch queries leads to a large number of memory accesses. Automatic classification takes advantage of item (1) by grouping like documents into cells which are segments of memory (tracks, cylinders, etc.) which do not require more than one memory access. Thus, it costs little extra in time to retrieve an entire cell than it would to retrieve a single document.

In addition, classification reduces the number of memory accesses required (item (2) above) by the very fact that the documents in a given cell are close to each other in content. This "likeness" increases the probability that multiple retrievals for a given query would appear in the same cell. This in turn reduces the number of cells accesses per query and hence the number of memory accesses required.

This reduction in memory accesses can be translated into greater on-line capacity for a system. Alternatively, it might speed operations up enough to justify slower, but less costly (see Table 2-3) mass storage devices.

2.3 Contributions of the Dissertation

The overall accomplishments of this dissertation are:

1) Definition of some of the problems involved in automated large-scale information storage and retrieval systems.
2) Provision of a superior method of solution for these problems.
3) Demonstration of the feasibility and advantages of this solution.

The methodology on which this solution is based is that of automatic classification of the document collection. Feasibility is demonstrated by automatically classifying a file of 50,000 document descriptions. The advantages of automatic classification are demonstrated by establishing methods for measuring the quality of classification systems and applying these measures to a number of different classification strategies. By indexing the 50,000 documents by independent methods, it is shown that these advantages are not dependent upon the indexing method used.

The advantages demonstrated are:

1) Automaticity.
2) Flexibility.
3) Browsability.
4) Reduction in storage space.

5) Reduction in retrieval time.

CHAPTER 3

LITERATURE REVIEW

3.1 Content of this Chapter

The main body of this chapter is concerned with a critical review of prior publications in the area of automatic classification theories and experiments. However, because of the importance of indexing and its close ties with classification, a descriptive section on indexing is included. Another reason for including a review of indexing efforts is that in preparing one of the indexes for the experimental file used for this dissertation, the author utilized a form of automatic indexing (see Appendix A).

3.2 General Critique of Prior Experiments

There is one item which has been remarkably constant in all research on automatic classifcation to date (the present study and the work leading up to it excluded). That is the lack of experiments on a significantly large data base. The largest corpus used for automatic classification experiments reported in the literature contains about one thousand documents (see Section 3.3). It is difficult to imagine how one could obtain a reasonable number of significant categories, much less a reasonable hierarchy, with so few documents (most experiments

were done on fewer tha 400 documents). The current experiments have clearly shown the need for large-scale experiments. For example, it has been found (reported in Chapter 5 of this paper) that in a certain aspect, based on 2500 documents, an inverted file outperforms a classified file. If the experiments stopped there the results would have been in error, for the classified file caught up to and far surpassed the performance of the inverted file as the number of documents processed were increased to almost 50,000.

Classification takes a number of different forms. As stated previously, most classification systems are a priori and have humans do the classifying. The newer of these systems generally use categories as indexes, thereby placing a document in more than one category. These systems have come into being in order to overcome the disadvantages of the Dewey decimal classification. However, they have only partially succeeded; and, in addition, have generally increased the notational complexity. Examples of such systems are the Colon Classification [6,107] and the Universal Decimal Classification [49,50, 52,53,54,115]. Reviews of these and other faceted classification schemes can be found in Vickery [146] and Taulbee [136].

In the realm of automatic classification, one can identify two levels of automation. One is the automatic

placement of documents into a priori categories. The other is the use of automated techniques to derive the classification categories (a posteriori) and then place the documents into these categories. Experiments have been performed on each of these levels, though it is felt that the latter is by far the more significant. The need for a good a posteriori automatic classification technique can be found in the literature. Altmann [2] notes that the potential users of a system under design demanded "browsability." In the absence of a decent automatic system, a new, a priori classification system was designed specifically for their collection. In designing a system for the Air Force, Debons, et al. [40] noted the limitations of all a priori classification systems and reluctantly decided that of the available systems, strict coordinate indexing, with no classification was the route to take. Lefkovitz, et al. [78,80,143] started using a posteriori automatic classification in a large-scale real-time chemical information system but abandoned it to inverted files due to the lack of data on the quality of automatic classification on large files.

Very few experiments have been performed on hierarchical a posteriori classification systems. A notable exception is the work of Doyle [42,43]. As stated previously, a hierarchy is required in order to have full browsing capabilities. In a 1964 review of the state of

the art of IS&R systems, Arnovich, et al. [4] did not consider the possibility of hierarchical, a posteriori classification.

A major obstacle to the development of any type of classification system is measurement of quality. Why design a system if there is no way to tell if it is better or worse than others? Until the present set of experiments, most classification systems were measured by one or more of the following methods:

(a) relevance assessments of documents in categories with respect to a few search requests,

(b) were documents placed into the same categories a human classifier would have placed them?,

(c) are a posteriori categories the same as a priori, human-organized categories?,

(d) do the categories "look good"? (subjective criterion).

Relevance assessment (a) of documents to requests should not be used in testing classification techniques. This use of relevance confuses classification with indexing. With the use of this measure, one cannot separate the quality of indexing from the quality of classifying and is more likely to be measuring the former. Secondly, the value of relevance and precision ratios being used in any of the ways they are today is open to question. A number of papers have been written pointing out the

shortcomings of the "pseudo-mathematical" use of these ratios [34,35,45,71,121,133,135].

Items (b) and (c) above assume some degree of a priori categories. In some experiments the categories are set up a posteriori on part of the collection and used to automatically classify the rest of the collection. In other cases the categories are set up a priori by humans. In either case, human judgment of "correct" document classification or "correct" category content is required. This is undesirable for two reasons. Firstly, humans are not terribly consistent in indexing or classifying documents [17, 93, 122]. Secondly, the goals of automatic classification are to serve the user as efficiently as possible and not to conform the system to preconceived ideas of categories of knowledge. It might be the case that these are one and the same, but this judgment must await experimental verification before it can be accepted.

Measure (d) above can be dismissed as vague, inconsistent, and not readily amenable to verification.

One attempt at an objective measure of classification can be found in Doyle [43]. Here, a collection of time-ordered items (daily work records and diaries) and portions of documents was classified. The criterion used was how well the categories could isolate continuous segments of time and tie together parts of the same documents.

One could not do direct extrapolation of these results to a more usual collection of documents, but it is a start towards objective measurements. Unfortunately, the collection consisted of only 100 items.

The measurement techniques used for this dissertation are explained later. However, for comparison with the above they will be briefly described here. These measures can be applied to any classification scheme (see Chapter 4 for five schemes to which they were applied) without human judgment. As described in Section 2.1, a count of the average number of discrete keywords per category yields a measure (to be minimized) of the "likeness" of documents in a category. In addition, if search requests are applied to a system (165 requests were used in this set of experiments), the number of categories looked at as well as the number of documents searched in these categories should be minimized. These measures can be used to compare the quality of two or more classification systems. In addition, the plot of keys per category versus number of categories could be thought of as an absolute measure, using the diagonal line of Fig. 2.1 as a reference line. However, this must be tempered by the fact that different collections probably have different relative minima for the average keys per cell. Therefore, for best measurements, experiments must be done on the same or similar files.

One of the reasons generally given for using few documents in an experiment is that automatic classification of large numbers of documents requires much time and/or high-speed memory (which can be converted into time via use of secondary storage) and that one or both of these factors are not available or cost too much to justify. Experiments are then performed with few documents but with full realization that actual systems would incur great expense in trying to classify large numbers of documents by the experimental methods [73]. In most automatic clas. fication systems being considered today (clustering types), classification time is proportional to the square, or even the cube, of the number of documents in the system. This is because of the need to compare every document (or partial category) with every other document (or partial category) or to generate and manipulate matrixes whose sides are proportiona. to the number of documents and/or the number of discrete keywords in the system (see Doyle [44], "Breaking the Cost Barrier in Automatic Classification"). This means that the cost of classification per document goes up at least linearly with the number of documents. Considering collections numbering in the millions of documents, it is evident that systems with the above characteristics are unacceptable.

There are two systems which are known to break

this N_d^2 effect. One is proposed (discussed on page 50) by Doyle [44] and the other is presented in this dissertation. Both are hierarchic, a posteriori classification systems which work from the top (all the documents) down (to the categories). In both, the time proportionality factor (N_d documents) is approximately $N_d \log N_d$, where the logarithmic base is the number of branches at each node of the hierarchy.

The cost involved with processing millions of documents eliminates the application of more sophisticated (i.e., theoretical) and/or deterministic techniques towards the problems of automatic classification. Few would doubt the possibilities of translating the problems of automatic classification into the realm of automata or linear programming theory. However, the practicalities of millions of different documents with tens of thousands of discrete keywords and tens of millions of keyword appearances eliminates the use of these otherwise attractive-looking devices.

Before the examination of particular automatic classification schemes, a word should be said about another type of classification. This is classifying, grouping, or relating index terms to aid retrieval but not to be used to group documents. These schemes could have merit in certain cases, but are not of direct interest here. Examples can be found in [45,56,74,75,79,114,130,154].

The most extensive of these studies was done by Zimmerman [154] who automatically classified the keywords in 25000 document descriptions into Exclusive and Inclusive groups. However, possibly due to the use of only 85 keywords, his results are not very encouraging for the future of this type of classification.

3.3 Automatic Classification

Table 3-1 (a-d) summarizes some significant facts about previous automatic classification experiments. This table and the cited references (plus the following discussion) are presented in lieu of a detailed description of each classification algorithm and experiment. Many reviews are available which describe the actual algorithms used to set up the categories or the statistical processes used to classify the documents [16,59,72,73,136,147], but none summarize the vital statistics for more than a very few experiments. Other, more general, reviews of automatic classification are also available [19,128,129] imbedded in reviews of broader areas of interest.

Lance and Williams [72,73] divide a posteriori classification strategies into hierarchical systems and clustering systems. They further subdivide hierarchical systems into agglomerative methods and divisive methods. In agglomerative methods (the only type they consider) the hierarchy is formed by combining documents, groups of

Source	Maron [84]	Borko [15]	Borko and Bernick [20]	Williams [150]
Year of Publication	1961	1962	1963-64	1963
Related Pubs.[1]	--	--	[21]	--
Corpus Size[2]	247/85	618	243/372	300/83
Number of Keywords	90	90	90	30?
Number of Groups	32	4, 10, 18	21	20
A Posteriori?	NO	YES	YES	NO
Hierarchical?	--	NO	NO	--
Name of Classification	--	Factor Analysis	Factor Analysis	Discriminant Method
Main Concept in Classification	statistical	matrix	matrix	statistical
Corpus Topic	Computers[3]	Psychology	Computers[3]	Computers[3]
Criterion[4]	(b)	(c)	(b)	(b)
Automatic Indexing?	YES	YES	YES	YES
Indexed Text	Abstracts	Abstracts	Abstracts	Abstracts

Table 3-1a

Automatic Classification Experiments

Source	Bonner [14]	Borko [17]	Doyle [42]	Dale and Dale [37]
Year of Publication	1964	1964	1964	1965
Related Pubs.[1]	--	--	[44,148]	[38]
Corpus Size[2]	350	659/997	100	260
Number of Keywords	18	150	?	90
Number of Groups	3, 4	11	1-100	8, 19
A Posteriori?	YES	YES	YES	YES
Hierarchical?	NO	NO	YES	NO
Name of Classification	Clustering	Factor Analysis	Ward grouping	Clumping
Main Concept in Classification	graphical	matrix	comparative	graphical
Corpus Topic	Medical symptoms	Psychology	German affairs	Computers[3]
Criterion[4]	(d)	(b)	(d)	(a)
Automatic Indexing?	NO	YES	NO	YES
Indexed Text	--	Abstracts	--	Abstracts

Table 3-1b

Automatic Classification Experiments

Source	Atherton and Borko [4]	Atherton and Borko [4]	Doyle [43]	Rocchio [110]
Year of Publication	1965	1965	1965	1966
Related Pubs.[1]	--	--	[148]	[57,82,116]
Corpus Size[2]	77	350	100	405
Number of Keywords	96	145	?	4041
Number of Groups	5	8	1-100	20, 30, 40
A Posteriori?	YES	YES	YES	YES
Hierarchical?	NO	NO	YES	NO
Name of Classification	Factor Analysis	Factor Analysis	Ward grouping	Clustering
Main Concept in Classification	matrix	matrix	comparative	graphical
Corpus Topic	Nuclear physicists	Nuclear	Time-ordered items	Computers[3]
Criterion[4]	(d)	(d)	time groups	(a)
Automatic Indexing?	?	?	NO	YES
Indexed Text	Description of work	Search requests	--	Abstracts

Table 3-1c

Automatic Classification Experiments

Source	Perriens and Williams [93]	Sparck Jones and Needham [127]
Year of Publication	1967	1968
Related Pubs.[1]	[151]	[88,89,90, 126]
Corpus Size[2]	1022/89,673	165
Number of Keywords	180	641
Number of Groups	6	?
A Posteriori?	NO	YES
Hierarchical?	--	NO
Name of Classification	Discriminant Method	Clumping
Main Concept in Classification	statistical	graphical
Corpus Topic	Broadcast items News articles	Aeronautics
Criterion[4]	(b)	(a)
Automatic Indexing?	YES	NO
Indexed Text	Items	--

NOTES

1. These expand the theory, present partial results, or are extensions of the described experiments.
2. Number of items actually used. Where A/B is shown;
   A = number of documents used to set up categories or to obtain criteria for a priori categories,
   B = number of documents automatically classified.
3. Used same basic corpus of 405 abstracts from IRE PGEC, March, June, and September, 1959.
4. See Section 3.2 for definitions of criteria (a), (b), (c), and (d).

Table 3-1d

Automatic Classification Experiments

documents, and groups of groups of documents until all documents are in one large group: the entire collection itself. The hierarchy being thus formed, all that remains is to select some criterion, such as category size, by which one cuts off the bottom of the hierarchy, thereby producing categories. Experiments using such a method were performed by Doyle [42,43] using the Ward grouping program [148]. Prywes [97,98,99] has also devised a system of this type. Wolfberg [153] has done some preliminary work on this algorithm, but because of computational difficulties with large collections, no large-scale experiments have yet been performed.

Divisive techniques have long been thought the realm of philosophers and other designers of a priori breakdowns of knowledge. With this technique, one starts with the entire collection and successively subdivides it until appropriately sized categories are obtained. Doyle [44] has proposed a system of this type (see Dattola [39] for preliminary experiments). However, this system requires some a priori categories as a starting point at each level of classification. Whether or not this can be overcome (it probably can) remains to be seen.

The algorithm used (among others) in this paper - called "CLASFY" - is also of the hierarchical divisive type, but is of a self-starting variety. Previous experiments performed during the development of CLASFY

are summarized in Table 3-2. For completeness, the present experiments using CLASFY are summarized in Table 3-3.

Clustering systems involve a wide variety of classification techniques which seek to group index terms or documents with high association factors together into "clusters", "clumps", or "factors" without trying to obtain a hierarchy. Some examples of experiments with some of these methods are shown in Table 3-1. Most of these methods require matrix manipulation, though it should be added that the precise manner of these manipulations varies widely with the particular scheme used. Another scheme of this general type is latent class analysis, proposed by Baker [7,8]. This method can utilize correlations of triplets or larger sets of index terms as well as pairs of terms as utilized by the other methods. There have been no reports of experiments using latent class analysis.

Price and Schiminovich [96] have recently manually simulated automatic clustering of 240 physics documents using bibliographic citations instead of keywords as the basis for the clustering algorithm. Even though this technique might be satisfactory under certain limited conditions, because of the variability of authors in citing references it is doubtful that this could be used as a general method. The quality criterion used in the above experiment was equivalent to (c) of Section 3.2.

Source	Angell [3]	Lefkovitz and Angell [77]
Year of Publication	1966	1966
Related Pubs.	[76]	[31]
Corpus Size	2500	4000
Number of Keywords	500	6000
Number of Groups	1-49	70
A Posteriori?	YES	YES
Hierarchical?	YES	YES
Name of Classification	CLASFY	CLASFY
Main Concept in Classification	multi-pass, comparative	multi-pass, comparative
Corpus Topic	(artificial surrogates)[5]	Aerospace
Criterion	description redundancy vs. random categories	none
Automatic Indexing?	--	NO
Indexed Text	--	--

NOTE

5. Produced by random number generation.

Table 3-2

Prior Experiments Using CLASFY

Source	Litofsky	Litofsky	Litofsky
Year of Publication	1969	1969	1969
Related Pubs.	--	--	--
Corpus Size	2254	46821	46942
Number of Keywords	2557	8044	13302
Number of Groups	1-216	1-1284	1-1183
A Posteriori?	YES	YES	YES
Hierarchical?	YES	YES	YES
Name of Classification	CLASFY	CLASFY	CLASFY - - - - -others were investigated but were found to be inferior to CLASFY
Main Concept in Classification	multi-pass, comparative	multi-pass, comparative	multi-pass, comparative
Corpus Topic	Nuclear	Nuclear	Nuclear
Criterion	Comparison with four other (plus variants) classification schemes (including a priori) for each file with respect to minimizing keywords in categories and, via 165 search requests, minimizing categories looked at and documents searched.		
Automatic Indexing?	NO	NO	YES
Indexed Text	--	--	Titles

Table 3-3

Summary of Present Experiments

Some other papers of interest to automatic classification are: O'Connor [91] (an "old" article on classification designed for peek-a-boo cards), Chien and Preparata [28] (file organization after automatic classification), Soergel [124] (highly theoretical - doubtful practical application), Nagy [87] (application of various automatic classification techniques to pattern recognition), and Sokal [125] (numerical taxonomy, or, the use of automatic techniques in biological classification).

3.4 Indexing and Automatic Indexing

Many more experiments have been carried out on indexing than on classification for automated IS&R systems. In addition, the indexing experiments were generally designed to be much more effective in selecting good indexing systems than were the classification experiments in selecting good classification systems. For example, out of 26 index evaluation projects reported in tabular form by Bourne [22], 21 of them involved comparative evaluations of from two to as many as about 15 different indexing schemes, some automatic and some not. Stevens [128] presents a good state-of-the-art report on automatic indexing and related problems as of early 1965. More recent reviews of automatic indexing are also available [9,19,108]. Henderson [59] recently gathered informative abstracts of a number of papers dealing with IS&R systems

with particular emphasis on those dealing with indexing.

Once again, a significant deficiency in most automatic indexing experiments is the small collections upon which conclusions are based. A notable exception to this are the experiments of Guiliano and Jones [55] where collections of 10,000 documents were used. Other deficiencies of automatic indexing experiments are similar to some of those (particularly the notion of relevance) described in Section 3.2 for automatic classification.

Since this paper is not directly concerned with indexing, specific indexing projects will not be reviewed. Instead, a few words will be said about comparative indexing experiments, especially those pertaining to the type of automatic indexing described in Appendix A.

Reasearch on comparative indexing has progressed from comparing manual indexes (such as Cleverdon, et al. [30]) to more recent work on comparing manual indexing with various forms of automatic indexing. A detailed analysis of various modes of automatic indexing is being performed by project SMART under the guidance of Salton [116,117,118,120]. Some of the items under study are document length (title, abstracts vs. full text), matching functions and term weights, language normalization (delete suffix "s", word stems, full thesaurus, etc.), manual indexing, and synonym and phrase recognition.

Salton has found that, in general, for his test collection: detailed manual indexing (over 30 terms per document) is slightly better than the automatic indexing techniques used, indexing on abstracts is better than on title alone, and use of a thesaurus involving synonym recognition is more effective than word stems which is slightly better than deleting only suffix "s" and common words.

Other experiments have been performed comparing indexing by title words vs. abstract words [68,109] and title words (usually KWIC, keyword-in-context) vs. manual indexes [1,12,24,29,69]. The general consensus is that titles of technical articles are sufficiently descriptive to be used for automatic indexing but that abstracts would probably serve somewhat better. These results (including those of project SMART) were used in semi-automaticlly indexing almost 50,000 documents (see Appendix A) used in the current experiments.

CHAPTER 4

EXPERIMENTAL CLASSIFICATION STRATEGIES

4.1 Introduction

This chapter contains descriptions of the various classification algorithms used in these experiments. The classification experiments themselves are described in the next chapter.

Five different algorithms were studied. Three of these are a posteriori, one a priori and one random, for comparison. Of the three a posteriori systems, only one is basically of a hierarchical nature (CLASFY). This system was studied with numerous variations of parameters and, as shall be seen, input orderings. It was found to be the best among those investigated. Because of the time required for processing of the large files, much of the parameter optimization was done on the small keyword file. In actual system operation the time required to classify a large file (assuming processing time increases no faster than $N_d \log N_d$ - see Chapter 3) is not so important because the classification would be performed once and not repeated until a substantial number of new documents have entered the system.

In all of the systems studied, a document acquired between classifications or memory reorganizations would be placed into a cell based on the original concept

of the particular classification system in question.

All CLASFY processing was done on an IBM 7040 computer in MAP. All other processing was done on an IBM 360/65 computer in PL/I.

4.2 A Hierarchical Classification Algorithm (CLASFY)

The original version of the primary classification algorithm under consideration here was conceived by Dr. David Lefkovitz [76*] and was programmed by Angell [3] in MAP on an IBM 7040 computer. Since then it has been used to automatically classify a document file for the Air Force [31,77]. In addition, it was used for a while in an experimental chemical IS&R system [78,80,143] but was discontinued because of lack of information about its performance on large files. However, until now, no means was available for measuring the quality of this algorithm. In the course of the current work, the above algorithm was improved and evaluated, and then compared with other classification systems.

4.2.1 Description of the Algorithm

CLASFY is a hierarchical classification algorithm of the divisive type. That is, one starts with the entire collection and successively partitions it into smaller and smaller groups until a group size criterion is met. These final groups are called cells and are the actual categories into which the documents are placed.

---

* Appendix B.

Each node is treated independently of the others. In other words, the algorithm is first applied to the entire collection. This results in partitioning the collection into N groups, represented by nodes in the classification tree. The selection of an appropriate node stratification number, N (i.e., number of branches out of a node or number of groups into which each node is partitioned) is discussed in Section 4.2.4.1. The algorithm is then reapplied to <u>each</u> of the resulting N nodes yielding N additional groups (assuming a constant stratification number) for each of these nodes. By now the collection has been divided into N^2 mutually exclusive groups of documents. This is continued until a size criterion, such as number of documents per group or number of computer words per group is met for each resulting group. Because collections are not completely homogeneous, the size criterion will generally be met at different tree levels for different portions of the classification tree. Therefore, in general, the resulting tree will <u>not</u> be a "regular" tree, terminating throughout at the same level.

Each node is represented by the keyword surrogates of the documents at that node and by the keyword vocabulary made up of the union of the keywords of these surrogates. The algorithm (operating at any given node) is based on three principles.

1) The keyword vocabulary is to be partitioned into some number of groups such that every

document description (of the documents at that node) is represented in at least one of the resulting groups.

2) The groups should be constructed such that each document description appears in as few groups as possible.

3) The number of keywords in each group should be roughly equal.

It should be noted that if principle 3) was not included, the solution to 1) and 2) would be to place all documents into one group. Of course, this would not result in any partitioning. The word "roughly" in principle 3) is executed by defining a sensitivity factor, E. An attempt is made to allow the number of keywords in each group to differ from those of any other group by no more than E (this can not always be done). Further discussion of the sensitivity factor can be found in Section 4.2.4.2.

Even though a document description may appear in more than one resulting group, it is assigned to only one of these groups (see the actual algorithm below).

The algorithm itself consists of a three pass process. That is, at each node, the keyword surrogates of the documents at that node are linearly scanned three times (actually, the third scan does not look at all the documents and, in fact, at times includes few or no documents).

Looking at the entire tree, this means that the entire collection of N_d documents is scanned linearly in three passes at each tree level. Since the time required for linear scans varies in proportion to the number of documents and the number of levels varies with the logarithm of the number of documents, it can be seen where $N_d \log N_d$ (where the logarithmic base is N, the stratification number) was arrived at for the proportionality factor for classification time (see Section 3.2).

In the following description of the classification algorithm for CLASFY, N represents the stratification Number, E the sensitivity factor, and D a particular document description (i.e., keyword surrogate). Figure 4-1 presents a macro-flowchart of CLASFY. For more detailed flowcharts of the original version of CLASFY, see Angell [3].

PASS 1

This pass partitions the keyword vocabulary of a node into N non-exclusive groups by adding the keywords of each document, one at a time, to one of the N groups.

1) Number the resulting groups 1,2,3...N. Initially, all groups have no keywords. The file is positioned at the beginning.
2) The next description, D, is read. Denote the group which contains the most keywords of D, group i. If there are two or more such groups,

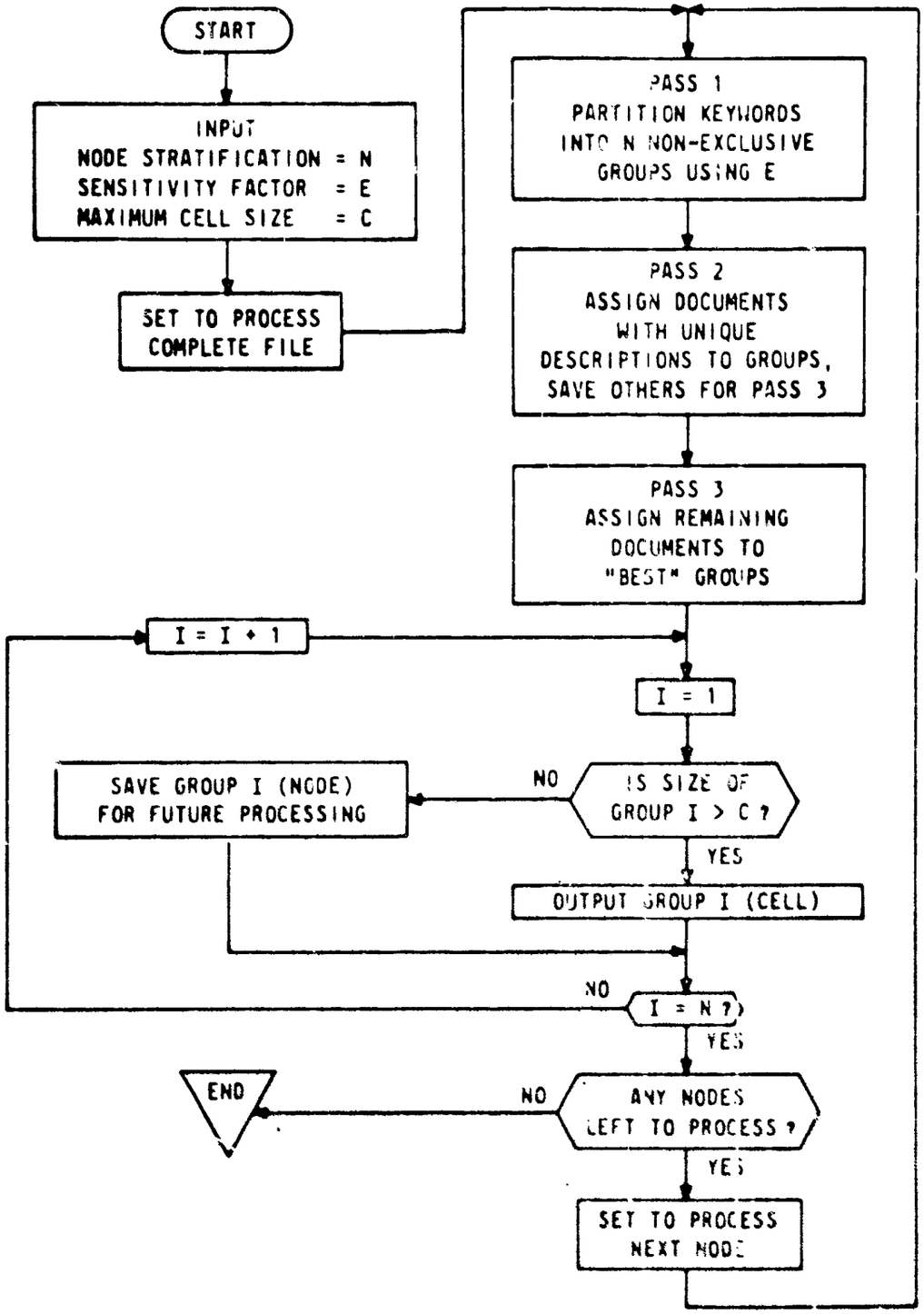


FIGURE 4-1

MACRO-FLOWCHART OF CLASSY

denote the one with the fewest distinct keywords as group i. If there are still two or more groups, arbitrarily select the one with the lowest group number as group i.

3) Let the number of keywords in group i be denoted n_i and the number of keywords of D <u>not</u> in group i (i.e., the number of keywords which would have to be added to group i if D were included in that group) be denoted as a_i. The following inequality is tested for $j = 1,2...N$, $j \neq i$:

$$(n_i + a_i) \leq (n_j + a_j) + E.$$

If true, that is, if the new size of group i is no more than E greater than the potential new size of any other group, the keywords of D are added (union) to the keywords of group i. Otherwise, set i = j (that j for which the above expression is not true) and continue the above test on the remainder of the groups.

4) If this is the last document, return to the beginning of the file and go on to Pass 2. If, not, go to item 2) of this pass.

It can be seen that this process guarantees that the keywords of every document description are included in <u>at least</u> one group. However, no documents have been assigned to any group.

PASS 2

This pass assigns those documents whose descriptions appear in only one group to that specific group. Documents with descriptions in more than one group are deferred for Pass 3 processing.

1) The next description, D, is read. If all the keywords of D appear in only one group, those keywords (of D) are flagged in that group and D is assigned to that group. The flagged keywords are essential because no other group contains all the keywords of D.

2) If the keywords of D appear in more than one group, no keywords are flagged and D is written on an intermediate file for Pass 3 processing. This indicates that a redundancy exists.

3) If this is the last document, position the intermediate file at the beginning and go on to Pass 3. If not, go to item 1) of this pass.

At this point, some of the documents have been assigned groups and some of the keywords in the groups have been flagged.

PASS 3

This pass of redundant descriptions from Pass 2 attempts to minimize description redundancies among the groups of keywords.

1) The next description, D, on the intermediate

file is read.

2) If the keywords of D are all flagged within at least one group, assign D to the first such group encountered (other methods can also be used to select which of these groups, if more than one, to which D should be assigned). If the keywords of D are not all flagged within any group, consider the groups which contain all the keywords of D. Of these, determine which one has the most keywords of D flagged (if more than one, arbitrarily choose the one with the lowest group number). Assign D to that group and flag the remainder of the keywords of D in that group.

3) If this is the last document, processing is complete for this node. If not, go to item 1) of this pass.

All documents have now been assigned groups. The Unflagged keywords in each group are redundant and are not contained in any document description in that group. These new nodes are now ready for repartitioning, if desired. When the cell criterion has been met by a particular group, it is considered to be a cell and the keys associated with that cell are the flagged keys of the group.

Figure 4-2 shows part of a classification via CLASFY of the large keyword file. Pertinent parts of the

```
CLASSIFICATION DATE   681017
TOTAL ITEMS           004267      NODE C00005
PARTITIONS  05, E IS  075, CELL SIZE C00460 ITEMS
KEYS                     1  TO  008041
PHASE 1 CELL SWITCHES  00107
PHASE 2 REDUNDANCY     01544

GROUP  01    NO. OF KEYS 01166    NO. OF ITEMS 00739    NODE 000022
GROUP  02    NO. OF KEYS 01021    NO. OF ITEMS 00658    NODE 000023
GROUP  03    NO. OF KEYS 01085    NO. OF ITEMS 01725    NODE 000024
GROUP  04    NO. OF KEYS 01070    NO. OF ITEMS 00752    NODE 000025
GROUP  05    NO. OF KEYS 01157    NO. OF ITEMS 00393    NODE 000026
TERMINAL NODE

DISTRIBUTION OF KEYS OVER GROUPS
GROUP(S)    KEYS
   05       00148
   04       00185
   03       00297
   02       00704
   01       01720
TOTAL       03054
```

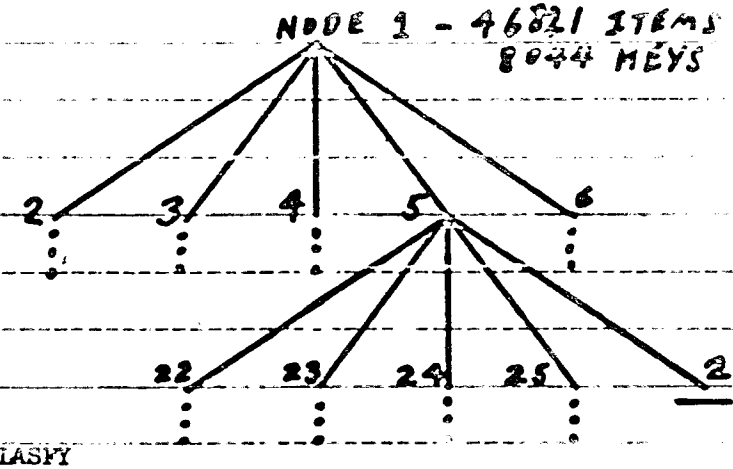


Figure 4-2

Part of Classification Using CLASFY

hierarchy are also shown. This particular node (number 5) contains 4267 documents (items) and N = 5, E = 75, and C = 460 documents. It should be noted that group 5 passes the size criterion (393 documents) and therefore is a terminal node (node 26), or cell.

4.2.2 Classification Example

The difficulty with presenting examples of classification is that systems such as CLASFY were designed for large numbers of documents. Their use on small collections will often produce poor classifications. With this in mind, the following example was chosen to illustrate aspects of the various classification algorithms, and not because "good" classifications will be produced.

A file of 14 document descriptions are displayed in Figure 4-3. The keywords of the documents were ordered and replaced by rank numbers as described in Section A.2. These integer keywords will be used as the document descriptors. This file will be classified with N = 2 and E = 0. The classification will be carried out on two levels, disregarding any cell criteria.

Figure 4-4 shows the three pass partitioning of the top node (14 documents). The keywords are shown in the order that they were added to the groups. Note that in Pass 3, documents D1 and D10 were added to group 1 and keywords 8 and 9 were found to be redundant and were

Document Name	Keywords	Integer Keywords
D1	A B	2 3
D2	A C D	2 4 5
D3	A C F	2 4 6
D4	A E F	2 7 6
D5	B D K	3 5 1
D6	B C	3 4
D7	D M N	5 10 11
D8	E K	7 1
D9	F G H	6 12 13
D10	I J	8 9
D11	J K	9 1
D12	I K	8 1
D13	K L	1 14
D14	M N O	10 11 15

Figure 4-3

A Sample File of Document Descriptions

Group	1	2
Keywords	2	2
	3	4
	7	5
	6	6
	5	3
	1	10
	12	11
	13	8
	9	9
	8	15
	14	

PASS 1

Group	1	2
Documents	D4	D2
	D5	D3
	D8	D6
	D9	D7
	D11	D14
	D12	
	D13	

Intermediate File
(redundant descriptions)

Documents D1
D10

PASS 2

Group	1	2
Keywords	2	2
	3	4
	7	5
	6	6
	5	3
	1	10
	12	11
	13	15
	9	
	8	
	14	

Documents	D1	D2
	D4	D3
	D5	D6
	D8	D7
	D9	D14
	D10	
	D11	
	D12	
	D13	

PASS 3

Figure 4-4

Partition of Top Level, N = 2, E = 0

deleted from group 2.

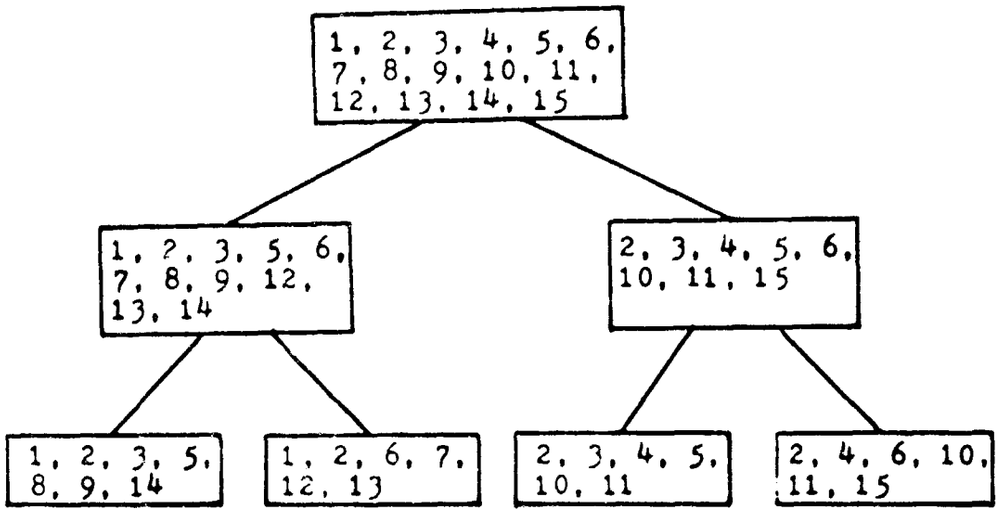
Each of the two groups of documents produced by the partitioning (see Pass 3, Fig. 4-4) are now successively partitioned to form a third level. The resulting three level binary tree is shown in Figure 4-5. The numbers in the boxes represent the keywords of the group which formed that node. This tree (but not the document groups) will be modified later on in this chapter in order to facilitate browsing.

If further partitioning were desired, some or all of the four groups of documents could be used as input to the next level partitioning. For example, if the maximum documents per cell was set at three, only Group I would be repartitioned, the rest becoming terminal nodes, or cells.

4.2.3 Unusual Situations

Two related situations which are not taken care of in the basic CLASFY algorithm were encountered in processing the large files.

The first can occur upon a proper combination of a relatively large sensitivity factor, relatively small number of documents (i.e., just above the cell criterion), and relatively small vocabulary of keywords contained in these documents. When these conditions allow, some of the N groups formed are empty. This occurs because, during



D1 -2 3
D5 -3 5 1
D10-8 9
D11-9 1
D12-8 1
D13-1 14

D4-2 7 6
D8-7 1
D9-6 12 13

D2-2 4 5
D6-3 4
D7-5 10 11

D3 -2 4 6
D14-10 11 15

Group I | Group II | Group III | Group IV

Average Keys per Cell = 6.25

Figure 4-5

Classification Tree, 14 Documents, N = 2, E = 0

Pass 1 processing, the relation

$$(n_i + a_i) \leq (n_j + a_j) + E$$

can hold for <u>all</u> the remaining documents even though one or more of the n_j (number of keywords in group j) are zero. When this situation arises, the empty groups are ignored and are not counted in the total number of cells (even though they trivially pass the cell critierion).

A more serious situation occurs when this aforementioned condition is carried to extremes. That is, when all but one group is empty. This is a rare situation but is most likely to occur (and has in both large files) when the documents of a node have few keywords per document and there is a high degree of overlap between keywords. The best example of this is a set of identical document descriptions numbering more than the cell criterion. When this situation occurs, the classification process is endless, for each partitioning will result in one group of all the documents and N - 1 empty groups. The solution arrived at is when this is recognized by encountering N-1 empty descriptor groups at the end of Pass 1, to arbitrarily partition the node into N equally sized (if possible) groups of documents.

4.2.4 Discussion of Parameters

4.2.4.1 Stratification Numbers

The stratification (also called ramification) number of a node is the number of branches (partitions) leading out of that node. When searching for an optimum value for this number, a number of factors must be taken into account. Among these are:

1) What is the best number in relation to user efficiency of browsing through a hierarchy?
2) What is the best number in relation to minimizing the number of keywords per cell for any given number of cells?
3) How does the size and scope of the collection affect the optimum stratification number?

Another point to consider is the advisability of a fixed stratification number versus a varying one. For simplicity, in these experiments the stratification number was selected at the start of each classification and was not changed. Of course, the lower the stratification number, the deeper (more levels) the classification tree will be.

With reference to item 1) above, Prywes, et al. [103,104] have found that based on minimizing decision time, the node stratification number has a broad optimum at e (2.718...) More recently, Thompson, et al. [138,139]

have taken "window shift time" as well as "decision time" into account. Thompson defines these terms as:

> "Decision time required for visually orienting to an alternative branch, focusing on the word or statement describing the alternative, and deciding whether or not the alternative is in the direction in which to continue the search."
>
> and "Window shift time required to shift the viewing window to the next level of the tree and visually to orient oneself to the new display."

In an on-line interactive system, window shift time is that time required to perform an indicating function, such as touching a point on a CRT display with a light pen, plus the time required for the computer to retrieve and display a new tree section. It was found that the optimum stratification number is dependent upon the ratio of these quantities, but independent of the size of the data base. For realistic estimates of this ratio, the optimum node stratification number was found to always lie in the range 3 - 5.

Classification experiments were performed on the small keyword file to answer item 2) above. These experiments varied the stratification number, N, while keeping the sensitivity factor E, constant. It was found that on going from N = 2 to N = 3, there was a reduction of about five percent in the number of keys per cell. However, increasing N beyond 3 did not significantly affect the number of keys per cell.

Based on the above and the intuitive feeling that

the answer to item 3) is that the stratification number should be somewhat larger for larger collections, the stratification numbers used for the major experiments of this research were chosen to be N = 3 for the small file and N = 5 for both large files. One possible justification for increasing N with collection size is that this slows the increase in classification time. In fact, if $\log_N N_d$ were kept a constant, the classification time would be proportional to N_d and hence the time (i.e., cost) per document would remain a constant for any collection size. If $\log_N N_d$ were set equal to seven ($\log_3 2254 = 7.03$, $\log_5 46900 = 6.64$ for the collections studied here), N would not reach ten until 10^7 documents were in the collection.

4.2.4.2 Sensitivity Factor

The sensitivity factor, E, strongly affects the quality of the final classification. E is used during Pass 1 to control the relative sizes of the groups of keywords as they grow. A small E tends to even out the number of keywords per group, while a large E tends to emphasize keyword co-occurrence among the document descriptions. For small numbers of documents, the number of documents per group is approximately proportional to the number of keys per group. Therefore, since it is desirable to form groups of about the same size, for small collections

(or the lower nodes of classifications of large collections) a relatively small value of E is desirable. This does not necessarily hold for large collections. For example. in Fig. 4-2, group 3 has 1085 keys and 1725 documents, while group 5 has 1157 keys and only 393 documents. A more extreme example is a case (in a classification of 46,821 documents) where two groups with the same number of keywords had 27387 and 2367 documents respectively.

The fact that larger values of E emphasize keyword co-occurrence, and hence, better classifications is illustrated in Figure 4-6. These curves (only parts of which are shown in the diagram) show that, holding all other parameters constant, fewer keys per cell (i.e., better classification) results from increasing E. However, the improvement gained by increasing E decreases as E gets larger. This can be seen in Fig. 4-6 by observing that the decreases in keys per cell are about equal for E going from 0 to 10, from 10 to 50 and from 50 to 150. The effect from increasing E in Pass 1 is felt in later passes by decreasing the number of redundant descriptions processed in Pass 3 ("PHASE 2 REDUNDANCY" of Fig. 4-2). For the examples shown in Fig. 4-6, the sum of the redundant descriptions processed by Pass 3 of the first three levels of the hierarchies are 2029 for E = 0 and 1267 for E = 50.

The net result is that E should be set as high as

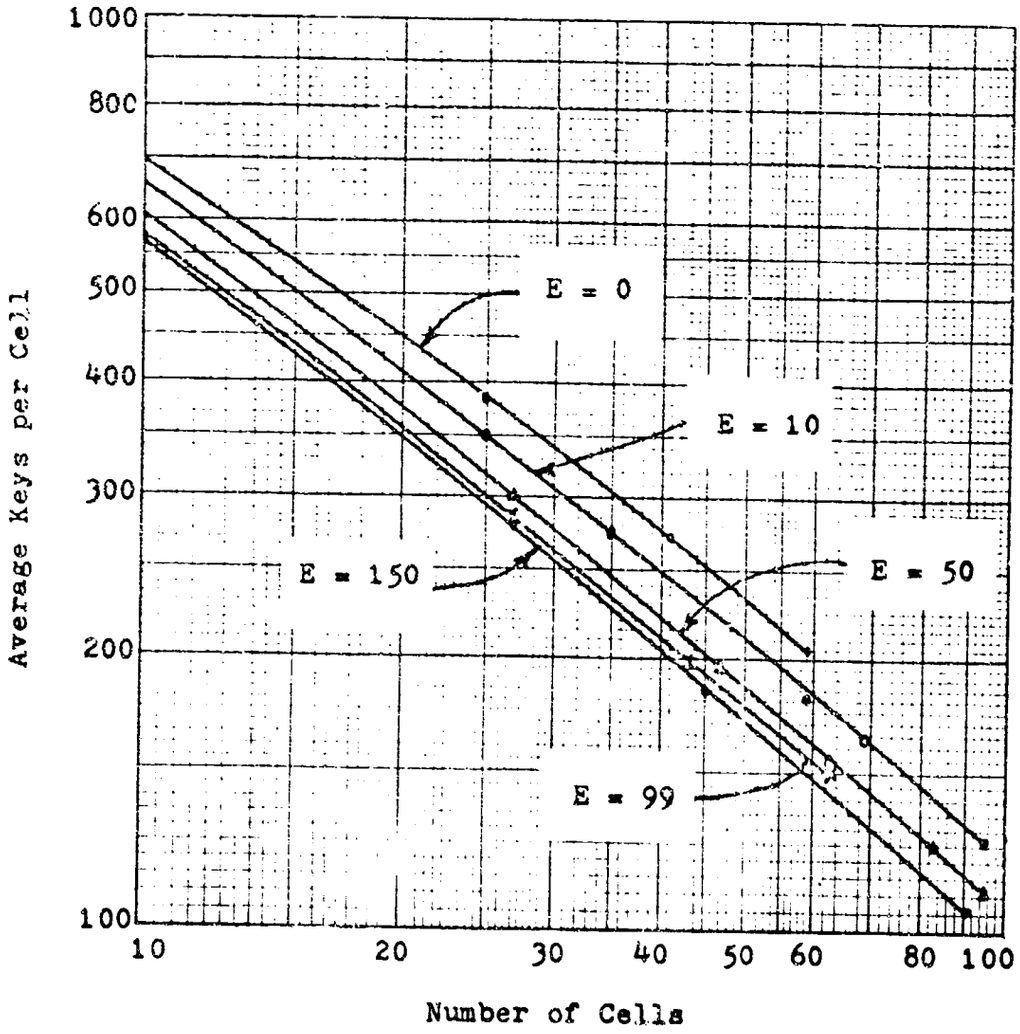


Figure 4-6

Effects of E on Keys per Cell, N = 3, N_d = 2254

possible without greatly unbalancing the size of the groups and hence, the structure of the hierarchy. With this in mind (for the experiments reported in chapter 5), E was set at 50 for the small file. In order to take advantage of all the above aspects of E, E was set at 75 - 150 for the top of the classifications of the large files and varied down to 25 - 40 for the lower nodes of the classifications.

4.2.5 Ordering of Input

The actual categories formed by CLASFY and therefore, the quality of the classification, depend, to some extent, on the order in which the documents are processed. It is desirable to obtain a unique ordering of documents which optimizes the classification. A number of different orderings were tried, some unique (independent of the original order) and some not (e.g., random ordering). One particular ordering was found to outperform all of the others for all three files.

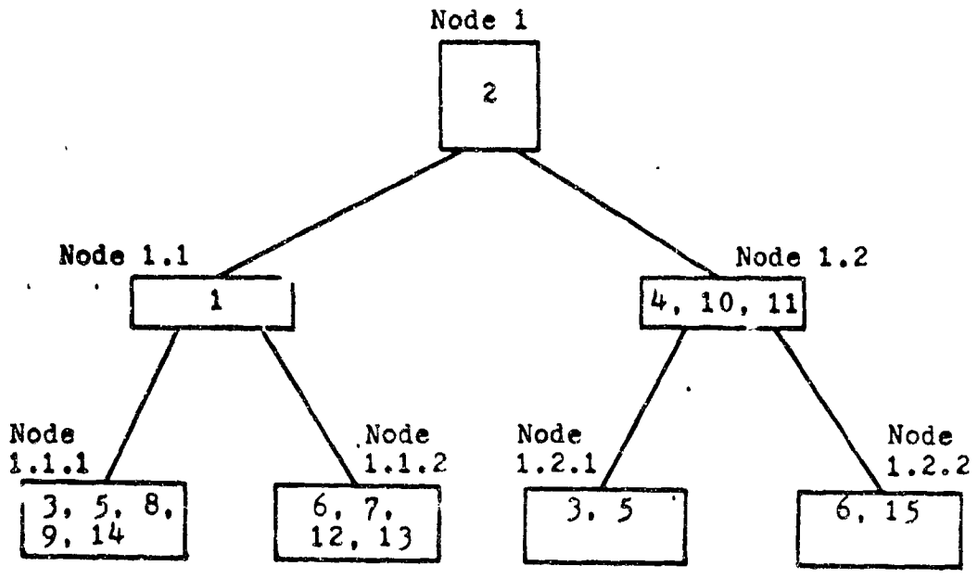
Because the orderings used are similar to basic elements of the other classification algorithms described in this chapter, they will not be discussed here. The reporting of the results of the different orderings will be deferred until Chapter 5.

4.3 Hierarchy Generation

The classification tree of Figure 4-5 does not present a hierarchy suitable for browsing. For browsing, it is desirable for the more general terms to be near the top of the hierarchy, progressing downwards until the most specific terms are near the bottom.

The hierarchy of keywords is formed from the bottom to the top. It should be noted that this keyword hierarchy is not used as a semantic hierarchy in a thesaurus in order to obtain descriptors for documents, but comes about a posteriori. Initially, the terminal nodes, or cells, are assigned the keywords which result from the union of the keyword surrogates of the documents in that cell. The keywords of the N terminal nodes under a parent (next level up the hierarchy) node are then intersected and those resulting keywords are assigned to the parent node. The keyword sets of the original N nodes are then deleted of the keywords assigned to the parent node. This process is continued until the top node is reached. A hierarchy was generated for the example of Section 4.2.2 and is shown in Figure 4-7. This should be compared with the classification tree of Figure 4-5.

Figure 4-7 also indicates the canonical node numbers for the seven nodes in the hierarchy. This method of node numbering allows one to immediately determine the location of a node from its number. For example, cell III



CELL I	CELL II	CELL III	CELL IV
D1 -2 3	D4-2 7 6	D2-2 4 5	D3 -2 4 6
D5 -3 5 1	D8-7 1	D6-3 4	D14-10 11 15
D10-8 9	D9 6 12 13	D7-5 10 11	
D11-9 1			
D12-8 1			
D13-1 14			

Figure 4-7

Keyword Hierarchy for Example of Section 4.2.2

of Fig. 4-7 is node 1.2.1. To find this node one starts at the top (1), takes the second branch from the left (1.2) and then the first branch from the left (1.2.1). The number of digits in a node's number indicates the level of the node in the hierarchy (here, cell III is on the third level).

It might seem that the more frequently a term is used, the higher it should be in the hierarchy. In general, this is true. However, equally important is how a term is used. A keyword which is high on the frequency rank by virtue of appearing in almost every document description of a few specialties would not rise very high in the hierarchy. On the other hand, a keyword with the same frequency of occurrence but with broader appeal might rise close to the top of the hierarchy. Incidently, there is nothing in the hierarchy generation algorithm to prevent the same keyword from appearing at more than one node. In fact, this occurs for the majority of the keywords.

Two properties of this type of keyword hierarchy are worth noting [31,77]. The first is that the keywords of each document description are wholly contained in the set of keywords consisting of the keywords at the nodes in the direct path from the top of the hierarchy to the terminal node which contains that document. In other words, referring to Fig. 4-7, one is guaranteed that the keywords of document D7 all occur in the union of the keywords of

nodes 1, 1.2, and 1.2.1. The second property is that each keyword will appear at most once in any given path from the top of the hierarchy to a terminal node. This means that if a keyword appears at node 1.2, it cannot appear at nodes 1, 1.2.1, or 1.2.2.

It would seem that if a reasonable number of keywords exist above the lower levels of a hierarchy, it means that the classification algorithm did not do a very good job of grouping like documents. To some expent this is true. That is, the better the classification, the fewer keywords above the lowest two or three levels of the hierarchy. For an ideal collection from a classification viewpoint (i.e., the cells form a mutually exclusive partition of the entire keyword vocabulary), there would be no keywords above the cell level. However, in any real collection there are a sufficient number of keywords generic to enough segments of the collection to form a reasonable hierarchy, regardless of how good a classification one achieves.

Various tables are required for using a hierarchy of this nature in an IS&R system.

4.3.1 Node-to-Key Table

The node-to-key table is of direct use in browsing. A user might enter a hierarchy at node 1 and successively decide to proceed to node 1.2 and then 1.2.1.

The retrieval system must be able to quickly retrieve the keywords appearing at any given node. This is done by entering the node-to-key table with a node number and coming out with the corresponding key numbers. Figure 4-8 shows the node-to-key table for the example of this chapter.

This table is actually the internal representation of a hierarchy in a computer memory.

4.3.2 Key-to-Node Table

The key-to-node table is used for retrievals by conjunctions and disjunctions of keywords. For example, consider the key-to-node table of the current example shown in Figure 4-9. This is actually an inverted file on nodes as opposed to the usual inverted file on documents.

Suppose that the following is a document request (the keywords have already been converted to integer form):

4 & (2 v 3 v 7).

After entry into the key-to-node table this is converted to:

..2 & (1 v 1.1.1 v 1.2.1 v 1.1.2)

Because each description must occur in the path between the top node and a document's cell, and because of the nature of canonical node numbering, only those conjuncts which do not disagree in any digits (a missing digit is not a disagreement) constitute valid search paths. In

NODE	KEY
1	2
1.1	1
1.1.1	3, 5, 6, 9, 14
1.1.2	8, 7, 12, 13
1.2	4, 10, 11
1.2.1	3, 5
1.2.2	6, 15

Figure 4-8

Node-to-Key Table for Example

KEY	NODE
1	1.1
2	1
3	1.1.1, 1.2.1
4	1.2
5	1.1.1, 1.2.1
6	1.1.2, 1.2.2
7	1.1.2
8	1.1.1
9	1.1.1
10	1.2
11	1.2
12	1.1.2
13	1.1.2
14	1.1.1
15	1.2.2

Figure 4-9

Key-to-Node Table for Example

this example 1.2 & 1.1.1 and 1.2 & 1.1.2 do not constitute valid paths as they differ in the second digit (i.e., 1.2 & 1.1.2 imply that keywords 2 and 7 do not appear in any document descriptions). On the other hand, 1.2 & 1 results in node 1.2 and 1.2 & 1.2.1 results in node 1.2.1.

The original request could have been to display these nodes (via the node-to-key table) in order to browse through the tree without having to start at the top. If this were the case, nodes 1.2 and 1.2.1 would be displayed. However, if the request was for the documents themselves (as it is), a third table must be consulted.

4.3.3 Terminal Node Table

The terminal node table is used for converting from a node number to one or more cell addresses. For this example, the terminal node table would look like:

Terminal Node	Cell
1.1.1	I
1.1.2	II
1.2.1	III
1.2.2	IV

Upon entering this table with a node number, one retrieves all cell locations whose terminal node numbers match the incoming node number through the level of the incoming node number. For this example, the node numbers under question are 1.2 and 1.2.1 (see previous section). For

1.2, the terminal node table indicates 1.2.1 and 1.2.2 as terminal nodes and therfore cell III and IV are indicated. For 1.2.1, cell III is indicated once again. Thus, cell III is to be searched for the keyword functions (4 & 2) and (4 & 3) and cell IV for (4 & 2). When this is done (see Fig. 4-7), documents D2, D3, and D6 are retrieved.

It should be noted that in this system, the indication to search a cell does not guarantee that any document descriptions exist in that cell which satisfy the search request. One purpose of classification is to maximize the probability that if a cell is searched, there will be documents there which satisfy the original request.

4.4 Forward and Reverse Classifications

When working on any relatively complex task, one often wonders: "Isn't there an easier way of doing this?" With that thought in mind, an attempt was made to solve the classification problem by sorting the document file and then partitioning it in a single pass.

The results of this endeavor are two classification schemes, herein called forward and reverse classification, which differ only on the sorting order and not on the partitioning algorithm. In addition, since the resulting orderings are unique (independent of the original ordering) they were also used as input to CLASFY (see Section 4.2.5).

The rationale behind sorting as a classification

algorithm is that in a file sorted on keywords, most documents will have at least one, and many times more, keyword in common with its neighbors. In addition, if documents are forced together by virtue of having a few particular keywords the same, the chances are good that other keyword co-occurrences will exist.

4.4.1 Forward Ordering

For the forward ordering, the keywords of each document are first sorted (this was done in the actual experiments using linear selection with exchange [64]) in ascending order. Figure 4-10a shows this for the 14 documents of Figure 4-3. After this has been done to all the document descriptions, the strings of keywords are temporarily considered to be individual, variable length, strings of digits. All keywords must be considered to be of the same length as the longest keyword. For this example, the string of D7 would be 051011.

The documents are then sorted (in the actual experiments, IBM system sort routines [65] were used) in ascending order of keyword strings. This is shown in Figure 4-10b. Thus, the entire file has been ordered by frequency of the keywords occurring in the document descriptions.

Figure 4-11 shows the first 45 documents of the title word file in forward order. The headings on this figure are: NDOC - order number of document, ABNO - digit

Document	Keywords
D1	2 3
D2	2 4 5
D3	2 4 6
D4	2 6 7
D5	1 3 5
D6	3 4
D7	5 10 11
D8	1 7
D9	6 12 13
D10	8 9
D11	1 9
D12	1 8
D13	1 14
D14	10 11 15

(a)

Document	Keywords
D5	1 3 5
D8	1 7
D12	1 8
D11	1 9
D13	1 14
D1	2 3
D2	2 4 5
D3	2 4 6
D4	2 6 7
D6	3 4
D7	5 10 11
D9	6 12 13
D10	8 9
D14	10 11 15

(b)

Figure 4-10

Forward Ordering Example

NDOC	ABNO	SECT	NSEL	KEYS										
1	127996	2033	1	1										
2	135990	2030	1	1										
3	101380	2033	1	1										
4	114240	2033	1	1										
5	126560	2862	2	1	2									
6	111261	3015	9	1	2	3	29	32	53	107	303	457		
7	143611	3013	9	1	2	3	32	36	51	58	90	223		
8	118356	3025	7	1	2	3	41	46	100	271				
9	117412	3665	5	1	2	3	41	78						
10	133019	3025	9	1	2	3	46	100	150	294	1796			
11	144925	3838	6	1	2	3	69	571	772					
12	121799	3870	6	1	2	3	73	244	1033					
13	106370	2650	7	1	2	3	112	2890	7844	11198				
14	125740	3870	6	1	2	3	120	244	2475					
15	118362	3035	7	1	2	3	468	486	3078	3241				
16	126932	3035	4	1	2	3	932							
17	142197	3310	13	1	2	4	6	20	36	37	82	109	430	479	783	855
18	122225	3650	8	1	2	4	7	73	238	381	6642					
19	124390	2832	8	1	2	4	30	36	420	561	820					
20	101729	2832	12	1	2	4	50	66	104	177	319	506	873	1068	3848	
21	107362	3310	6	1	2	4	82	401	892							
22	136712	2862	8	1	2	4	97	273	443	548	1109					
23	119958	2510	11	1	2	5	9	22	28	72	111	148	207	756		
24	136346	2612	7	1	2	5	9	111	157	756						
25	143285	2620	11	1	2	5	10	15	66	104	111	136	533	1056		
26	141845	3230	6	1	2	5	30	125	304							
27	143647	3015	10	1	2	5	32	41	46	51	212	307	514			
28	113291	3025	8	1	2	5	100	312	344	412	543					
29	105925	2033	6	1	2	6	35	57	758							
30	122938	3025	10	1	2	6	36	104	121	170	271	345	999			
31	109198	3015	7	1	2	6	41	53	56	615						
32	115773	3858	7	1	2	6	93	100	153	304						
33	144185	3310	12	1	2	7	9	30	36	49	82	127	249	298	783	
34	116500	2862	9	1	2	7	9	49	50	59	127	991				
35	136477	2650	9	1	2	7	9	67	112	546	909	13039				
36	124940	3221	9	1	2	7	13	17	43	147	241	389				
37	134827	2880	10	1	2	7	30	129	166	295	689	1245	9479			
38	146146	3224	6	1	2	7	33	67	6759							
39	124598	2862	7	1	2	7	50	188	296	531						
40	145760	2854	6	1	2	7	56	490	517							
41	128588	2862	7	1	2	7	97	143	581	1592						
42	120315	2862	7	1	2	7	97	143	581	1592						
43	106737	2880	7	1	2	7	366	371	699	2138						
44	113060	2870	8		2	7	597	1849	2180	9232	11703					
45	104746	2650	5		2	7	1018	13021								

Figure 4-11

Title Word File, Forward Order

"1" followed by abstract (document) number (00000 - 47055), SECT - a priori category ("section") number (not used here - see Section 4.6), NSEL - number of keywords, KEYS - integer keywords.

It should be noted that this ordering does not always force "like" documents to be near each other. For example, consider two documents with descriptions 1 5 6 7 and 5 6 7 respectively. They would not be placed near each other because in the forward ordering, 5 6 7 would be placed after all the documents with keywords 1, 2, 3, and 4 while 1 5 6 7 would be placed at or near the beginning of the ordering.

4.4.2 Reverse Ordering

Basically, the reverse ordering is the opposite of the forward ordering. However, there is one very important difference. Without this difference, the order of the keywords in each document description would be switched. For example, 10 11 15 would become 15 11 10. However, 15 is a unique (i.e., occurs only once in the collection) keyword. Therefore, because 15 cannot occur in any other document, it doesn't make sense to use 15 as the highest order keyword for sorting. Therefore, only the order of the non-unique keywords is reversed in going from the forward to the reverse ordering. Now 10 11 15 becomes 11 10 15. The result of this keyword ordering for the example is

shown in Figure 4-12a (here, 12 is the lowest unique keyword).

The sort of the documents on the keyword strings proceeds as in the forward ordering except that the documents are now sorted in descending order of keyword strings. This is shown in Figure 4-12b.

4.4.3 Modified Orderings

These orderings can be improved to some extent. Consider the forward ordering of Fig 4-10b. The keywords of document D7 do not appear elsewhere in the vicinity of that document. This is because the other occurrences of keyword 5 occurred in descriptions with lower keywords (D2 and D5), and therefore were already accounted for. The logical location for D7 is next to D14 since both documents have keyword 10 (and, incidentally, keyword 11).

This situation can occur whenever all but one occurrence of a keyword appear in documents with lower (for forward ordering) or higher (but non-unique, for reverse ordering) keywords. This occurs in documents D6, D7, D9, D10, and D14 in the forward ordering (Fig. 4-10b) and documents D12, D6, D1, and D13 in the reverse ordering (Fig. 4-12b). While keyword co-occurrence cannot always be improved by modifying the ordering, it was shown by experimentation that it can in enough instances to make it worthwhile.

Document	Keywords
D1	3 2
D2	5 4 2
D3	6 4 2
D4	7 6 2
D5	5 3 1
D6	4 3
D7	11 10 5
D8	7 1
D9	6 12 13
D10	9 8
D11	9 1
D12	8 1
D13	1 14
D14	11 10 15

(a)

Document	Keywords
D14	11 10 15
D7	11 10 5
D10	9 8
D11	9 1
D12	8 1
D4	7 6 2
D8	7 1
D9	6 12 13
D3	6 4 2
D2	5 4 2
D5	5 3 1
D6	4 3
D1	3 2
D13	1 14

(b)

Figure 4-.2

Reverse Ordering Example

Figure 4-13 presents a flowchart for a program which modifies the ordering of the forward ordering. For the reverse ordering, one only has to change the statements involving CURRENT in boxes A, B, and C to: CURRENT = highest keyword (box A), CURRENT = CURRENT - 1 (Box B), and CURRENT = 1? (box C).

Figure 4-14 shows the resulting forward and reverse ordering for the file of the example after processing by the modification program. For the forward ordering, four documents (D6, D7, D9, and D10) were removed from the input file for later processing (i.e., were written onto the TEMP or LAST files) and three of these ultimately wound up in the LAST file (D6, D9, D10) before being outputted. Note that the LAST file collects all the documents whose keywords are unlike the first keyword of any other document in the final ordering (sort of a garbage collector). The reverse ordering had only two documents removed from the input file (D12 and D6) and none of these wound up in the LAST file. It should be noted that is is possible for a document to be rewritten many times on the TEMP file before being outputted or written onto the LAST file (and later outputted). The numbers of documents removed from the input for later processing and those written on the LAST file for the two orderings of each of the experimental files are shown in Table 4-1. Other statistics for these files can be found in Section A-4.

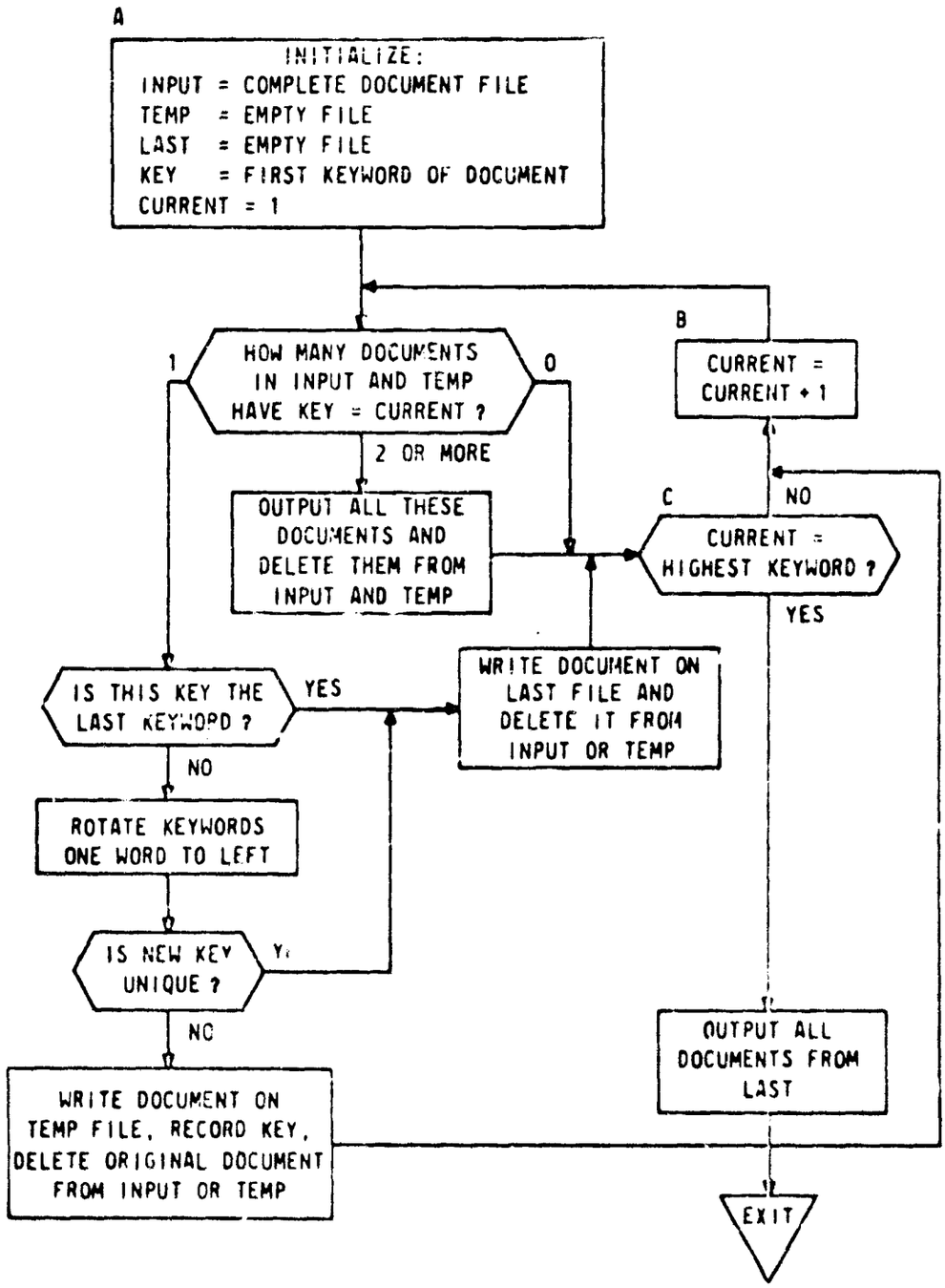


FIGURE 4-13

ORDER MODIFICATION PROGRAM (FORWARD ORDER)

Documents	Keywords
D5	1 3 5
D8	1 7
D12	1 8
D11	1 9
D13	1 14
D1	2 3
D2	2 4 5
D3	2 4 6
D4	2 6 7
D7	10 11 5
D14	10 11 15
D6	4 3
D9	12 13 6
D10	9 8

(a) Forward

Documents	Keywords
D14	11 10 15
D7	11 10 5
D10	9 8
D11	9 1
D4	7 6 2
D8	7 1
D9	6 12 13
D3	6 4 2
D2	5 4 2
D5	5 3 1
D6	3 4
D1	3 2
D13	1 14
D12	1 8

(b) Reverse

Figure 4-14

Final (Modified) Orderings

	Small K yword File		Large Keyword File		Title Word File	
	Forward	Reverse	Forward	Reverse	Forward	Reverse
Total documents	2254	2254	46821	46821	46942	46942
Documents removed from input for later processing	42	304	77	318	373	347
Documents written on IAST file	25	3	32	3	257	23

Table 4-1

Ordering Modification Statistics

Henceforth, reference to forward and reverse orderings will imply those orderings after modification.

4.4.4 Classification Algorithm

Probably the simplest method of transforming an ordering into a classification is, for N_d documents and N_c cells, to declare every N_d/N_c documents to be a cell. However, this would not make optimum use of the properties of an ordering. For example, consider the forward ordering of Fig. 4-14a. For four cells, $N_d/N_c = 3.5$. This would indicate that the first three or four documents should comprise a cell. However, it is obvious that, because each contains the keyword "1", the first five documents should be in the first cell.

An algorithm to allow for cases such as this has been programmed and is flowcharted in Figure 4-15. Two parameters are necessary: AVR, the projected average cell size ($\approx N_d/N_c$) and MAX, the maximum allowable cell size. The actual number of cells resulting from this classification procedure is not known in advance (the same is true for CLASFY) but is a function of the values chosen for AVR, MAX and the contents of the document file itself.

This algorithm tries to divide the documents into cells at points where the first keyword changes, but after AVR documents have been included. If this is not possible because the number of occurrences of a keyword in the

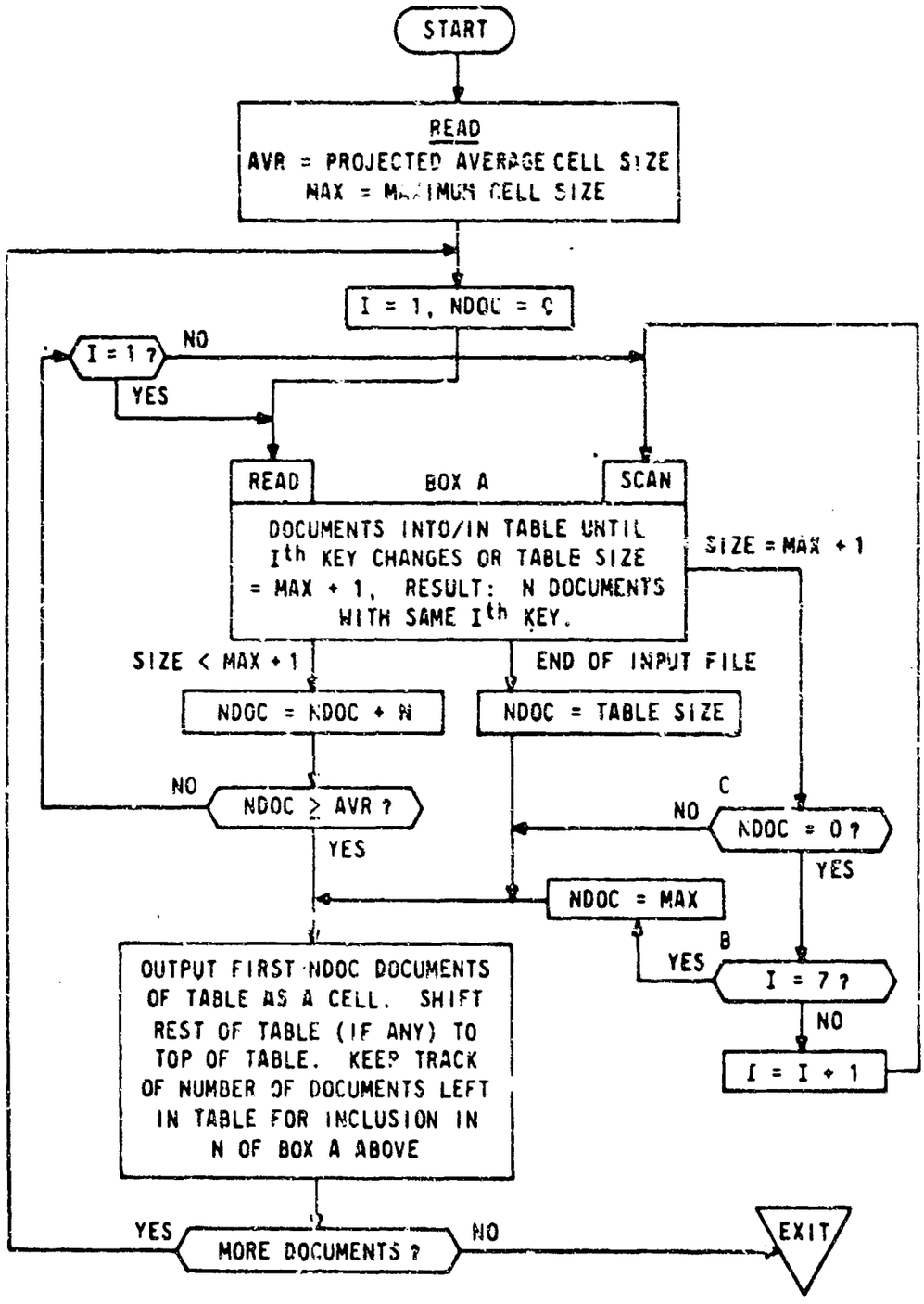


FIGURE 4-15

ORDERED FILE CLASSIFICATION ALGORITHM

first position in documents is larger than the value set for MAX (such as keyword "1" in the forward ordering of a large file, e.g., 4294 times for the large keyword file - see Fig. A-2), this process is attempted with the second keyword. This is contined until a keyword position is found where the cell division can take place. Since this would never occur if there are over MAX documents with identical descriptions, after a number of keyword positions have been investigated (here, arbitrarily set at seven - see box B of Fig. 4-15), the next MAX documents are considered to be a cell.

Even though an occasional cell will have fewer than AVR documents (i.e., if there are fewer than AVR first positional appearances of a keyword but the inclusion of the next keyword exceeds MAX documents - see the no branch out of box C in Fig. 4-15), most will have between AVR and MAX documents. In addition, given the same AVR and MAX, the average cell in a forward classification will be larger (hence fewer cells) than the average cell in a reverse classification because of the longer "runs" of first position keywords in the forward ordering.

Thus, AVR and MAX must be carefully chosen to obtain the number of cells desired (N_c) and, at the same time, optimize the classification. The more cells desired, the higher AVR must be set above N_d/N_c. It was found that for the experimental files, MAX should be set at about

four times AVR for best results for the forward classification. The same factor was used for the reverse classification but the largest cells formed were about two times AVR, therefore setting MAX above this value had no effect.

As an example, the following are statistics on classifying the large keyword file with a goal of about 250 cells (average cell size = 187 documents).

	Forward	Reverse
Number of documents, N_d	46821	46821
AVR set at	93	179
MAX set at	372	895
N_d/AVR = projected number of cells	503	262
Actual number of cells	283	247
Largest cell	369	314
Average documents per cell	165	190

Figure 4-16 shows the file of the example used in this chapter classified into four cells by both the forward and reverse classifications.

4.5 Random "Classification"

In order to see how each of the classification algorithms compared to doing nothing at all, a random "classification" (it is admittedly somewhat of a contradiction calling a random process a classification) was set up.

	File		
	Small Keyword	Large Keyword	Title Word
Number of documents, N_d	2254	46821	46942
Number of discrete keywords, N_v	2557	8044	13309
Indexing method	manual with thesaurus	manual with thesaurus	semi-automatic on title
Total keyword occurrences	19,262	466,810	277,141
Average keywords per document, N_{kd}	8.54	9.96	5.90
Average documents per keyword (i.e., average number of keyword occurrences)	7.53	58.03	20.82
Number of unique keywords (i.e., occur only once)	992	2189	5879
Total documents retrieved, 165 requests	862	27635	18753
Average retrievals per request	5.2	167.5	113.7

Table 5-1

File Statistics

This was done by assigning a random number [61] to each document and then sorting the file on this random number. The result of this is a randomly ordered file. One might expect something of this nature in a file ordered by accession number only. The file was then "classified" by considering every N_d/N_c documents to be a cell. This results in N_c cells for a file of N_d documents.

Besides using the random classification as a measure of the other classification systems, the random ordering described above was used as an additional file ordering for input to CLASFY.

4.6 Human (A Priori) Classification

The last of the five (CLASFY, forward, reverse, random, human) classification systems under study here is a manually-generated, a priori classification system. Each document in the data files used for these experiments (see Appendix A) has been manually assigned one of almost 300 categories. Examples of category numbers assigned to documents can be found as the four digit numbers under the "SECT" heading in Fig. 4-11.

These categories form a hierarchy of up to five levels (including the complete file as a level) with the following stratification numbers:

Node Level	Node Stratification
1	11
2	3 - 11
3	2 - 9
4	6 - 8

Because of the hierarchical nature of this system, documents in categories whose numbers differ slightly are supposed to be fairly close in content. This enables one to transform this classification into one of any number of categories.

Once again the files were sorted, but this time on category number. The files were then classified into $\approx N_c$ cells by dividing the files after every $\approx N_d/N_c$ documents, ensuring that the division (if possible) occurs at the end of a category. At should be noted that this is the same classification algorithm as that used for the forward and reverse classifications (see Section 4.4.4) with the category numbers considered as single keywords (i.e., one per document).

This classification was used to compare the quality of a posteriori automatic classification systems to that of a manually-generated a priori system.

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1 Data Files

Appendix A contains complete descriptions of the data files used for these experiments. The three files under study are the small keyword file, the large keyword file, and the title word file. The file statistics descussed in Section A.4 are repeated for convenience in Table 5-1 along with the retrieval statistics reported in Section B.2.

The small keyword file was used to set the parameters for the experiments on the large files. In addition, it was used in conjunction with the large keyword file in order to relate the classification results to file size (the large file has twenty times more documents than the small file). The most significant experimental results obtained here involve the large keyword file and the (large) title word file. These files contain essentially the same documents ($\approx$ 46900 each out of the same 47055 documents) but are indexed by independent methods, one manual and one automatic. The indexing of the same document collection by two different methods was done in order to determine if the quality of the various classification schemes is a function of the type of indexing used. Similar results from the two large files would therefore indicate that the

	File		
	Small Keyword	Large Keyword	Title Word
Number of documents, N_d	2254	46821	46942
Number of discrete keywords, N_v	2557	8044	13309
Indexing method	manual with thesaurus	manual with thesaurus	semi-automatic on title
Total keyword occurrences	19,262	466,810	277,141
Average keywords per document, N_{kd}	8.54	9.96	5.90
Average documents per keyword (i.e., average number of keyword occurrences)	7.53	58.03	20.82
Number of unique keywords (i.e., occur only once)	992	2189	5879
Total documents retrieved, 165 requests	862	27635	18753
Average retrievals per request	5.2	167.5	113.7

Table 5-1

File Statistics

	File		
	Small Keyword	Large Keyword	Title Word
Number of documents, N_d	2254	46821	46942
Number of discrete keywords, N_v	2557	8044	13309
Indexing method	manual with thesaurus	manual with thesaurus	semi-automatic on title
Total keyword occurrences	19,262	466,810	277,141
Average keywords per document, N_{kd}	8.54	9.96	5.90
Average documents per keyword (i.e., average number of keyword occurrences)	7.53	58.03	20.82
Number of unique keywords (i.e., occur only once)	992	2189	5879
Total documents retrieved, 165 requests	862	27635	18753
Average retrievals per request	5.2	167.5	113.7

Table 5-1

File Statistics

classification techniques used are relatively independent of indexing method. Later in this chapter it will be shown that this is indeed the case.

5.2 Experimental Measures of Quality of Classification

Various experiments were performed on the data files by themselves and on the files in conjunction with retrieval requests (see Appendix B and Table 5-1) in order to measure the relative quality of the classification schemes described in the previous chapter. The measures used are discussed below in this section, while the actual experiments and results are described in the following sections.

One objective measure of the quality of classification systems was discussed in Section 2.1. Documents in a cell are probably close in subject if there are a large number of keyword co-occurrences in that cell. Therefore, the average number of discrete keywords per cell (N_{kc}), with an aim towards minimization of this quantity, is one measure of the relative quality of classification systems.

There is a possibility, however remote, that a classification system can do fairly well on the above test and yet produce a poor set of categories from a retrieval efficiency point of view (i.e., does not sufficiently reduce the number of memory accesses per search request - see Section 2.2.5). This <u>might</u> come about if a classifi-

cation system happened to do a good job of bringing together documents with keywords which are generally not used in search requests, but a poor job of doing so with frequently (from a search request viewpoint) used keywords. Admittedly, it does not seem likely that a system with a very low average key per cell count could be very inefficient in actual memory accesses, but two systems of equal N_{KC} might differ in retrieval efficiency. In addition, the number of memory accesses required for a classified file can be compared to those required for an inverted file system. In this way, one can obtain a measure of the retrieval time savings offerred by automatic classification.

In order to test this, actual search requests can be applied to all classifications in question. The number of cells which must be searched per question is a measure of the number of memory accesses required per question. Naturally, the goal is to minimize this quantity. The number of retrieval requests used must be large enough for the results of this test to be significant.

Another measure is the number of documents searched per request. While it is true that this quantity should be dependent upon the number of cells searched, because of the variability in the size of cells additional insight can be obtained by measuring this quantity. If the number of cells searched per request is identical for two classification systems, then the better system is probably the one which

calls for searching fewer documents per request.

It is more difficult to quantitatively measure the quality of a hierarchy than it is to measure the quality of the classification itself. One measure of limited value is the number of entries in the key-to-node table (which is the same as the number of entries in the node-to-key table). A smaller key-to-node table indicates that more keywords have migrated upwards from the cells towards the apex of the tree, thereby producing a hierarchy richer in keywords, and requiring less storage space. Therefore, given two hierarchical systems with all else being equal, the one with the smaller key-to-node table is probably better.

The best means of measuring the quality of a hierarchy is probably large-scale testing of its usefulness for browsing (not done here). Short of this however, the alternative unfortunately is a subjective measure. This is to look at the keywords at the various nodes and decide whether they represent a distinct part of the collection or are merely a jumble of unrelated keywords. In a good hierarchy, one should be able to consider the keywords at a node as an abstract of the knowledge contained beneath that node in the tree.

As stated in the beginning of this section, all of these measures were used in the following experiments.

5.3 Keywords per Cell

Each file was classified numerous times by the various classification algorithms described in the previous chapter. All three files were classified using the following classification schemes: human, forward, reverse, random, and CLASFY with the file in the reverse order. In addition, both keyword files were classified by CLASFY with the files in forward and random orders.

The results of these classifications are shown in Figures 5-1, 5-2, and 5-3 for the small keyword, large keyword, and title word files, respectively. The results are presented in the form described in Section 2.1. That section can be referred to for the significance of the parallelogram envelopes and the various numbers on the axes.

Each classification of each file is plotted from one cell to N_d cells (i.e., one document per cell). The probable ranges of interest for the small and large files are as follows:

File Size	Documents	Cell Range of interest	Documents per Cell
Small	2254	30 - 125	75 - 18
Large	≈46900	200 - 1500	235 - 31

Because of the difficulty of displaying so many curves on single sheets of paper, the actual classification points are not shown. However, enough classifications were

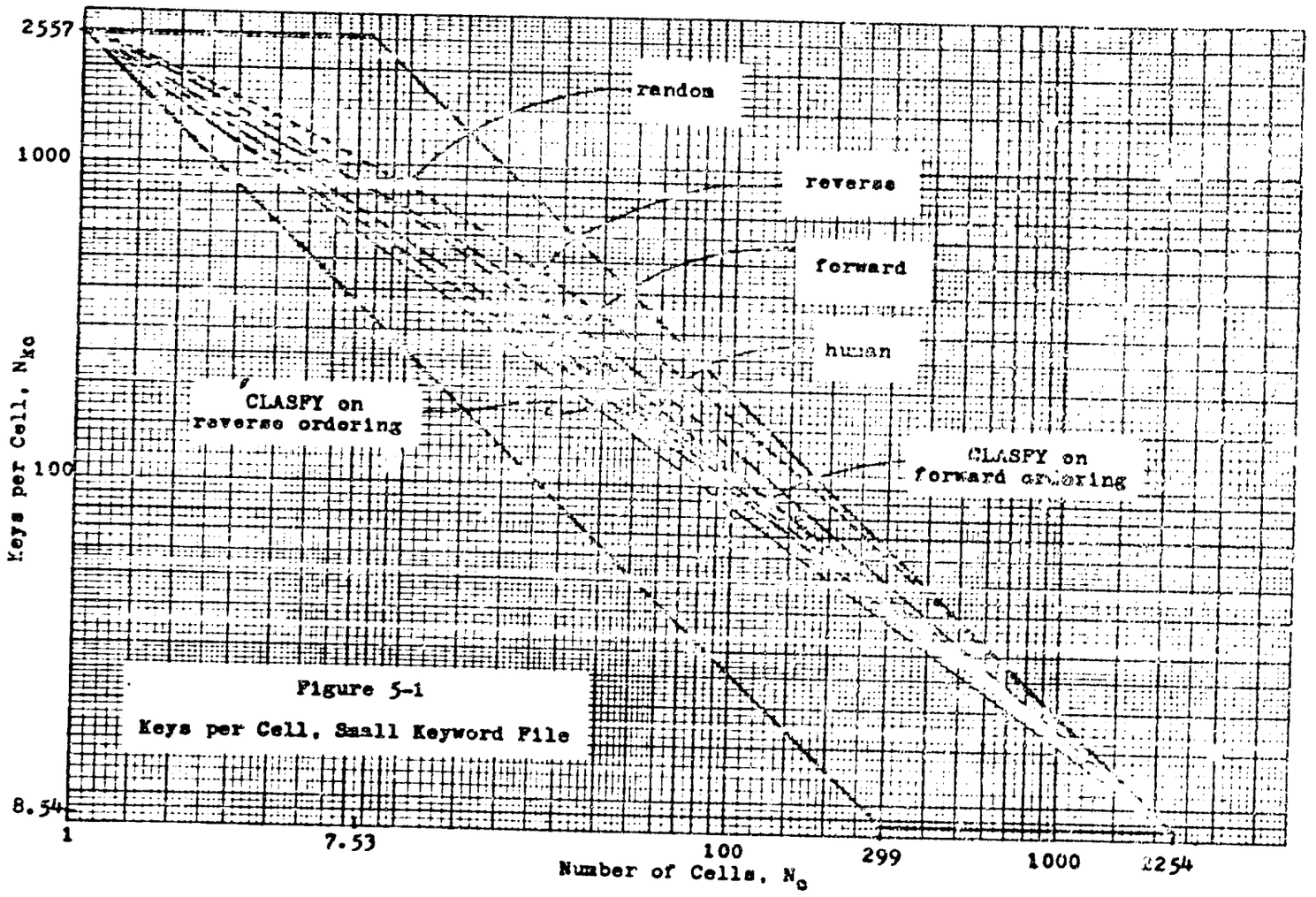


Figure 5-1

Keys per Cell, Small Keyword File

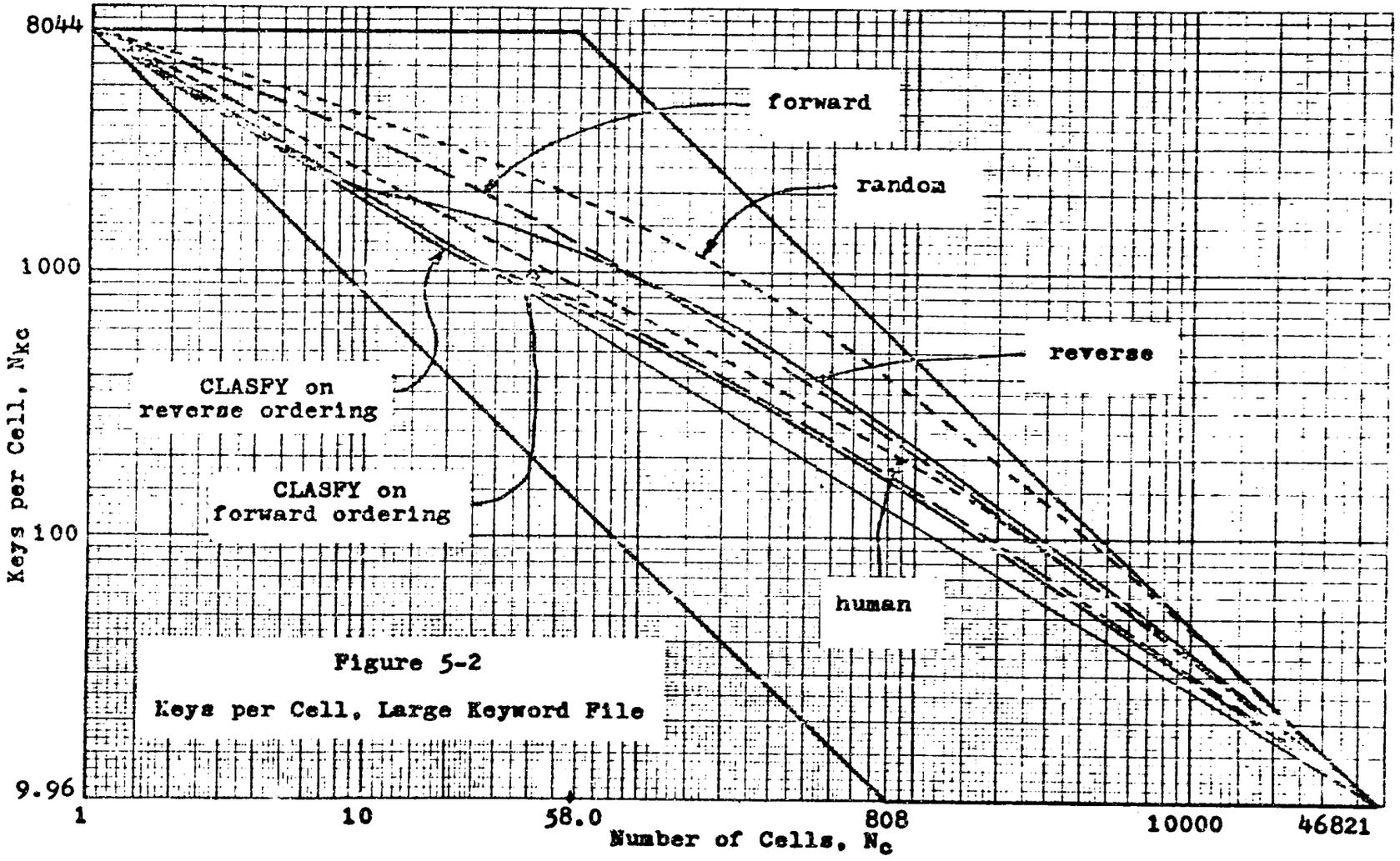


Figure 5-2

Keys per Cell, Large Keyword File

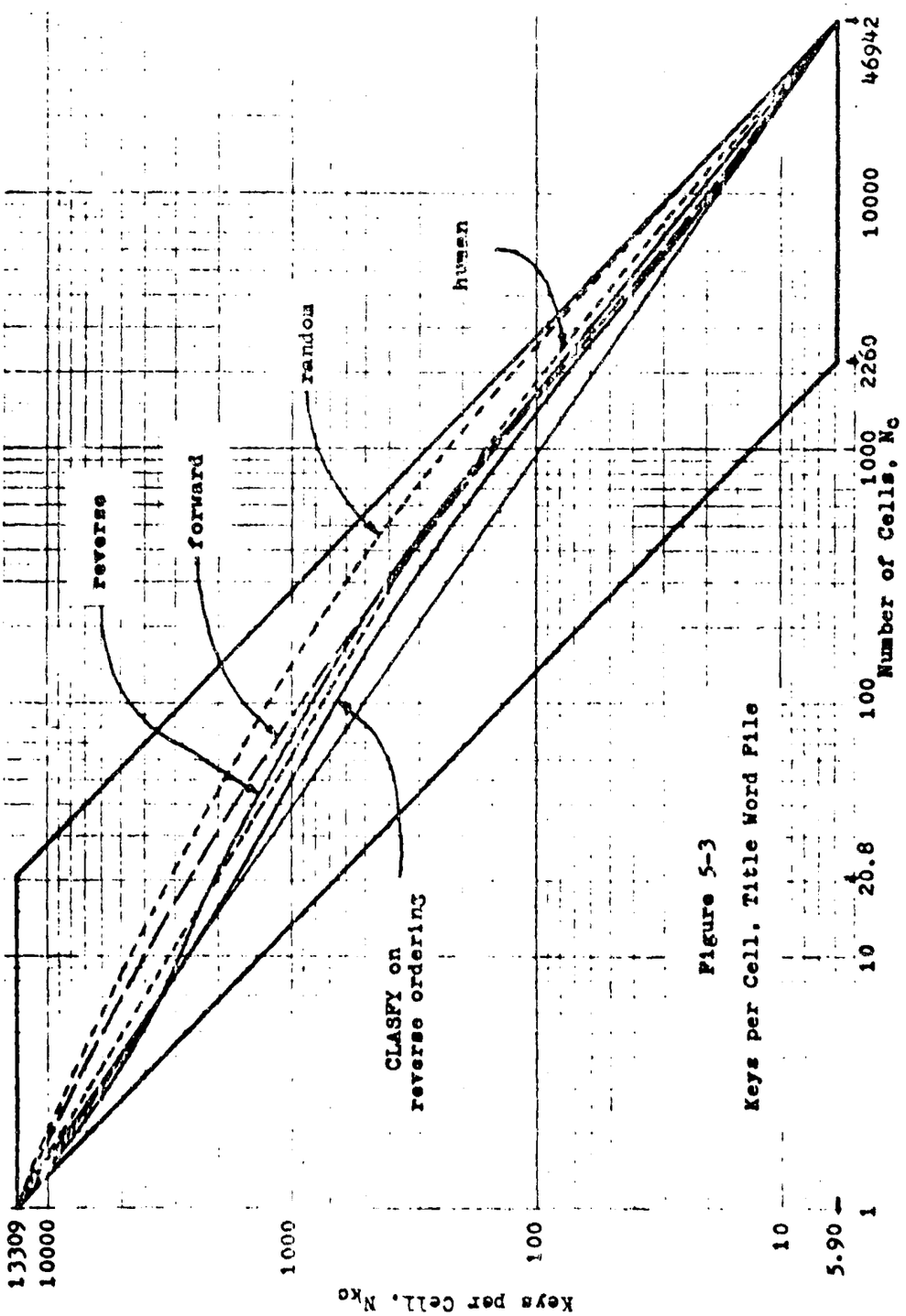


Figure 5-3

Keys per Cell, Title Word File

made to ensure smooth curves. The greatest number of cells produced for the large files was 2500 for all classifications but CLASFY. The most cells produced by CLASFY (large keyword file, reverse ordering) was 1284, requiring 5 hours and 20 minutes on an IBM 7040 computer. This time could be greatly reduced by the execution of CLASFY on a more modern computer (the other classifications were performed on an IBM 360/65) with faster cycle time and, of much greater importance, faster and larger capacity secondary storage facilities. However, since a file is not classified very often, as long as the classification time stays within reason (see Section 3.2) the actual processing time is not very significant.

The curves for CLASFY with the files in random order (keyword files only) were not drawn on Figures 5-1 and 5-2 because they lie completely between the curves of CLASFY with the files in forward and reverse orders.

The following are some observations and conclusions based upon Figures 5-1, 2, and 3.

1) The results for the three files are very similar. This implies that, based on the keyword files, the relative quality of the classification systems studied is independent of the size of the collection. In addition, based on the two large files, the implication is that the rela-

tive quality of the classification systems is also independent of how a collection is indexed. This leads one to the conclusion that, assuming the collection used here is representative of other technical collections, the order of quality of these systems (with some minor exceptions) is absolute and hence independent of the collection to be classified.

2) As expected, all the "legitimate" classification systems outperform the random classification by a considerable margin.

3) CLASFY is the best (based on the number of keys per cell) classification system studied. In particular, for all three files and for any number of cells, CLASFY outperforms the human, a priori system.

4) CLASFY outperforms the other systems regardless of the order of the file presented to the CLASFY algorithm. However, it performs best (by a small margin which decreases with increasing file size) when the input file is in reverse order.

5) The forward classification does poorly on few cells but improves and overtakes the human classification as the number of cells increases. This crossover in quality takes place in or just above (i.e., more cells than) the region of

interest.

6) The reverse classification starts off very well (few cells) but is not as good as the forward classification during most of and beyond the region of interest.

7) Even though CLASFY represents the best system studied here, its curves of N_{kc} vs. N_c are above the diagonals of the parellelogram. In Section 2.1 it was stated that these lines represent the approximate regions of the expected plots of N_{kc} vs. N_c. That statement was made based on these results and the realization that while CLASFY might be a good classification system, future study and experimentation will probably turn up better classification algorithms. In addition, just how close this plot is to the diagonal line is somewhat dependent upon the document collection as well as the classification system.

5.4 Results of Retrieval Requests

One hundred sixty-five actual retrieval requests were used to interrogate the files. See Appendix B for details on these requests. Table 5-1 shows the number of documents retrieved in each file as a result of these requests.

The requests were submitted to each classified file individually as if they were part of an on-line system. That is, batching of requests was not allowed. The number of cells searched and number of documents searched were recorded for each request and then totaled for the 165 requests. It should be noted that browsing was not used on the files classified by CLASFY or humans (the only hierarchic systems, and therefore the only ones which allow for browsing) to reduce the number of cells searched. However, browsing would probably be an integral part of any actual on-line system using CLASFY.

5.4.1 Theoretical Results with Inverted File

No experiments were performed with any of the files organized as an inverted file. However, because the documents of an inverted file are essentially in random order, it is possible to obtain some theoretical results.

The number of cells searched is a measure of the retrieval efficiency if a cell is equivalent to an appropriate unit of memory (see Section 2.2.5). Considering this to be the case, one can consider an inverted file as being divided linearly into physical cells. One can now calculate the average number of cells which must be entered per request for an inverted file.

The number and size of the cells will be considered to be the same as those of the classified files so that

the results will be directly comparable. However, there are two somewhat offsetting items which should be kept in mind. An inverted file is usually set up such that either of the following is true.

1) The records of the file are not blocked. This means that in response to a request, only the desired document need be transmitted into high-speed storage. Therefore, the input/output time required for this method, per memory access, is lower than that of a classified file. However, this organization requires more storage space than (2) below and hence more physical cells are required to contain the file. This results in more memory accesses per request.
2) The records in the file are blocked. This requires fewer cells than the above and hence fewer memory accesses; however, because a significant portion of a cell must be read into high-speed storage, the input/output time per memory access is not much less than that for a classified file.

In either of the above cases, the number of documents searched for an inverted file is equal to the number of documents retrieved.

The problem, therefore, is given X documents

(i.e., X documents will be retrieved) randomly distributed over N_c cells, what is the expected value for the number of non-empty cells. A constraint on the problem is that there is a maximum number of documents, C, which can be placed into each cell.

Consider a single cell. The probability of a particular document being placed in that cell, assuming completely random placement of documents, is $1/N_c$. The probability that i documents out of X are placed in that cell is therefore

$$\binom{X}{i}(1/N_c)^i(1 - 1/N_c)^{X-i}.$$

Since there are N_c cells, the expected (i.e., average) number of cells with i documents is

$$(N_c)\binom{X}{i}(1/N_c)^i(1 - 1/N_c)^{X-i}.$$

The number of cells with at least one but not more than C documents is what is desired. Hence, summing from i = 1 to C results in (up to now C has been ignored):

$$N_c \sum_{i=1}^{a} \binom{X}{i}(1/N_c)^i(1 - 1/N_c)^{X-i}$$

cells, where $a = \min(C,X)$.

It is noted that the above summation represents part of the cumulative binomial probability function. For cases when $C < X$ (i.e., $a = C$), N_c is large and hence $1/N_c$ is small. Looking up values for

$$\sum_{i=0}^{N} \binom{N}{i}p^i(1 - p)^{N-i}$$

under the condition of small values for p [188], one finds that this quantity is insignificant. Therefore, the

expected number of non-empty cells is approximately

$$N_c \sum_{i=1}^{X} \binom{X}{i} (1/N_c)^i (1 - 1/N_c)^{X-i}$$

cells. However, since by the binomial formula, for $p < 1$

$$\sum_{i=0}^{N} \binom{N}{i} p^i (1 - p)^{N-i} = 1,$$

the number of non-empty cells is equal to

$$N_c[1 - \binom{X}{0}(1/N_c)^0(1 - 1/N_c)^{X-0}] = N_c[1 - (1 - 1/N_c)^X].$$

This expression represents the number of cells which must be accessed for a request of X documents on a file divided into N_c cells. The above expression was evaluated for the parameters of the files and retrieval requests under study and is presented in graphical form, along with the classification results, in the next section.

5.4.2 Cells Searched

The number of cells searched when the 165 requests were applied to the various classifications of the small keyword file is shown in Figure 5-4. The results shown in that figure are not very encouraging because not only does the inverted file system cause fewer cells to be searched (accessed) than any of the other systems, but all the classification systems require more cells to be searched in the range of interest (i.e., 30 - 125 cells) than the number of documents retrieved! This is due to cells containing

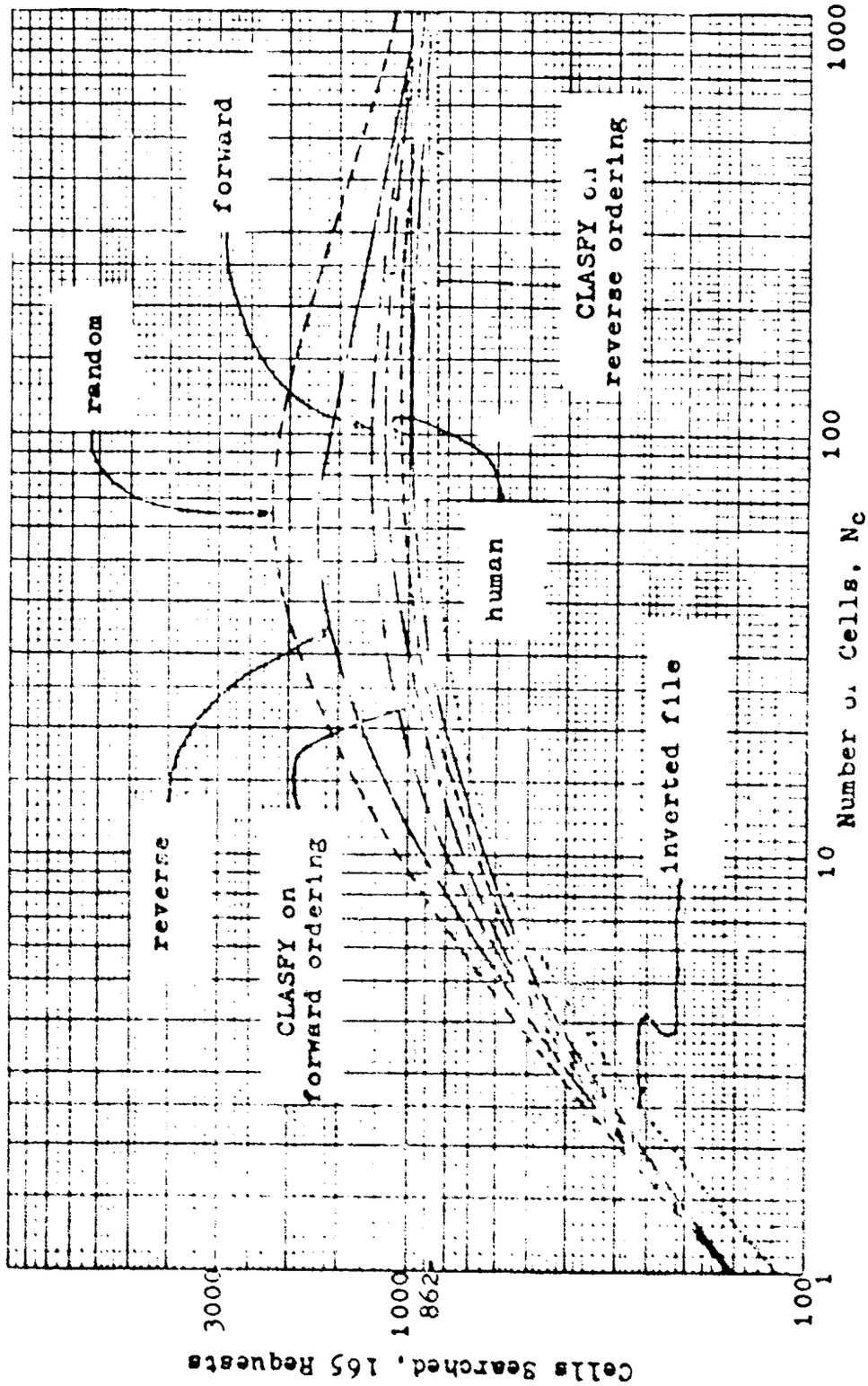


Figure 5-4
Cells Searched, Small Keyword File

keywords which match a request, but not having any documents in them which have the proper keywords to satisfy the request.

This is the problem with insufficient numbers of documents referred to in Section 3.2. By the time the number of documents have risen to almost 50000 (large keyword file), the situation has reversed itself. Figure 5-5 shows the cells searched vs. number of cells plots for the large keyword file. Here, the CLASFY (any ordering), human, and forward classifications surpass the inverted file. The same holds true for the title word file (Figure 5-6).

For the cells searched on the large keyword file, CLASFY with the input file in reverse order is best, barely edging out other input orderings to CLASFY and the human classification, with no others being close in the range of interest (200 - 1500 cells). In the case of the title word file, the CLASFY, human, and forward classification systems are all fairly close in the range of interest, in fact their plots cross over between 450 and 600 cells. It should be noted that for both large files, the number of cells searched for the reverse and random classification still exceed those searched for the inverted file and also exceed the number of documents retrieved.

A point made in Section 5.2 is illustrated in the

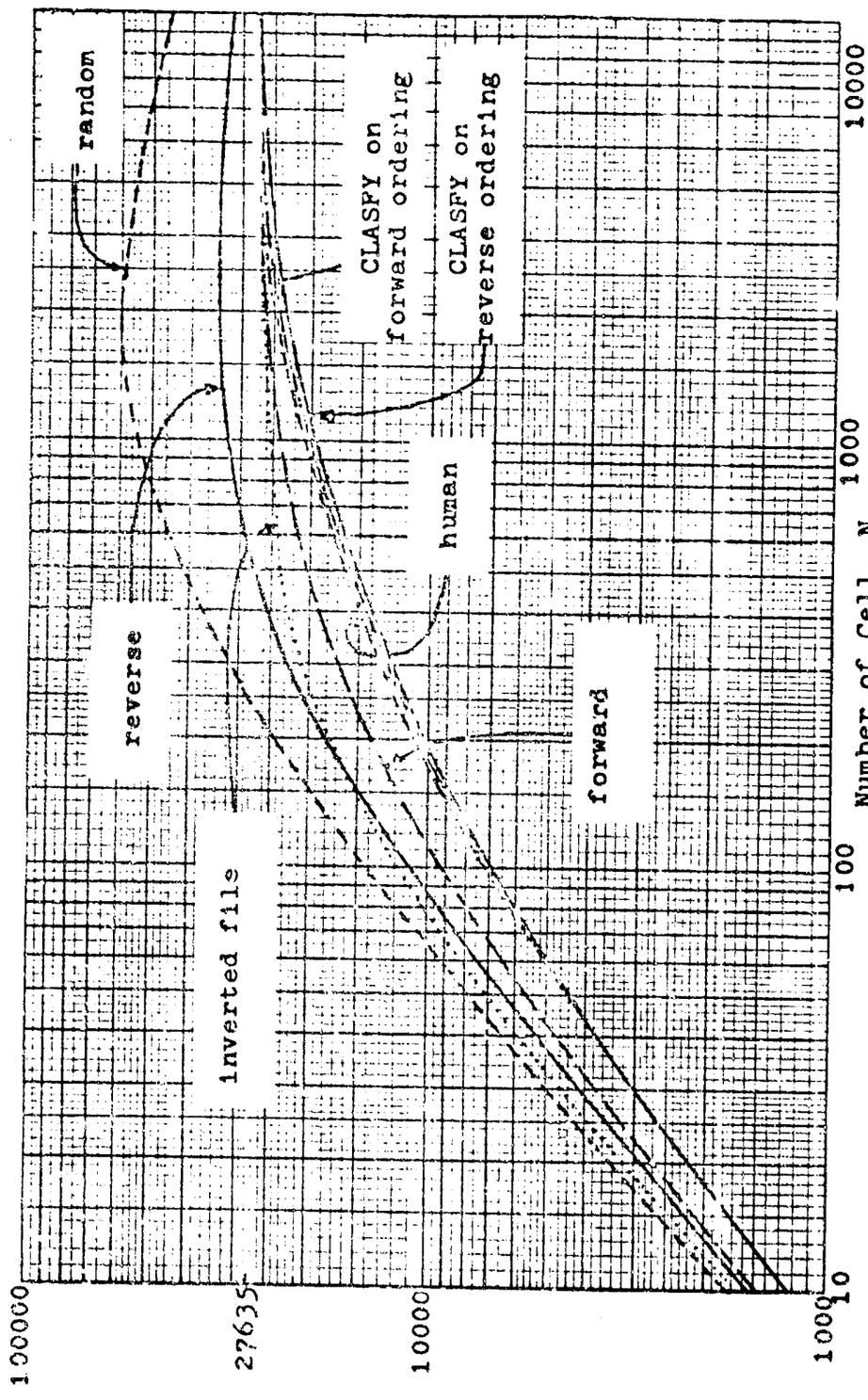


Figure 5-5

Cells Searched, Large Keyword File

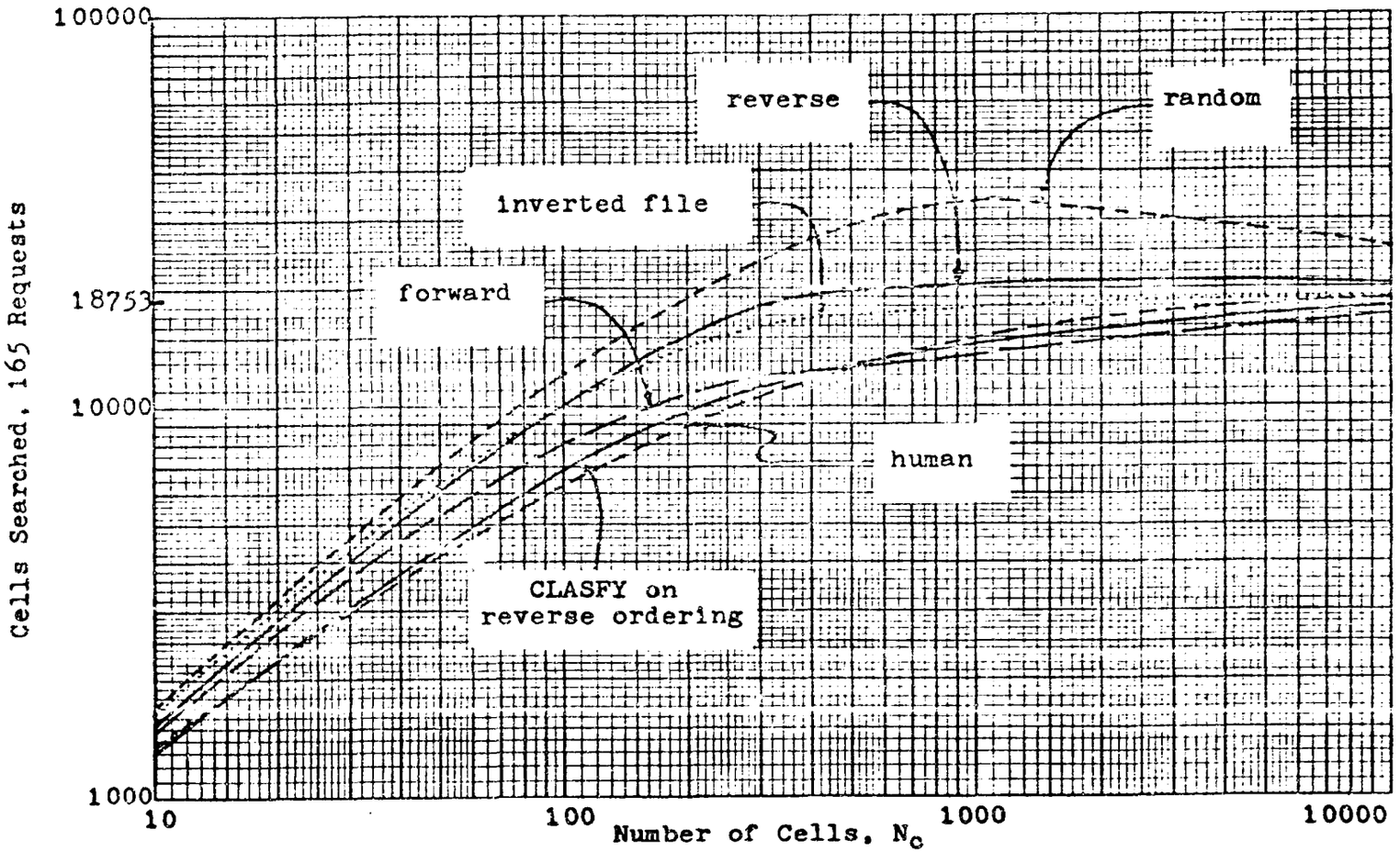


Figure 5-6

Cells Searched, Title Word File

graphs of the two measures discussed thus far of the title word file (Figures 5-3 and 5-6). On the basis of keys per cell, CLASFY is better than the forward classification for any number of cells. However, on the basis of cells searched, the reverse holds for more than 450 cells. This, and other examples which can be found in these two sets of graphs, shows that the different classification systems do indeed emphasize the co-occurrence of different keywords.

In the range of interest (200 - 1500 cells), the inverted file requires 19 to 78 percent more cells searches, and hence memory accesses, than does CLASFY for the large keyword file and 18 to 44 percent more than CLASFY for the title word file. In an on-line, large scale system the advantages of CLASFY should increase, perhaps drastically, for two reasons:

1) In going from the small to large keyword files the number of cells searched using CLASFY decreased tremendously relative to those searched for the inverted file. This trend can be expected to continue for larger files.
2) The number of cells searched shown in these figures does not take into account the browsing capabilities of CLASFY. The procedures followed for browsing in a hierarchy are described in Section 2.2.3. During the course of request refinement allowed by browsing, certain sections

of the tree would probably be eliminated because of lack of relevance to the retrieval request. This would reduce the number of cells (and documents) searched without appreciably decreasing the number of pertinent documents retrieved. This cannot be done in a non-hierarchic system, such as an inverted file. The magnitude of the advantage described above is potentially quite large; however, quantitative results are not available because appropriate experiments have not as yet been performed.

5.4.3 Documents Searched

The number of documents searched for a request is not completely dependent upon the number of cells searched. Due to the variations in the cell sizes, a different number of documents may be searched for the same number of cells searched.

This effect can be seen in the plots of the numbers of documents searched vs. number of cells shown in Figures 5-7, 5-8, and 5-9. It is noted that fewer documents are searched when the file is organized by the human classification than by any other system even though CLASFY outperforms the human classification (in most cases) in number of cells searched (Figures 5-4, 5-5, and 5-6). Part of

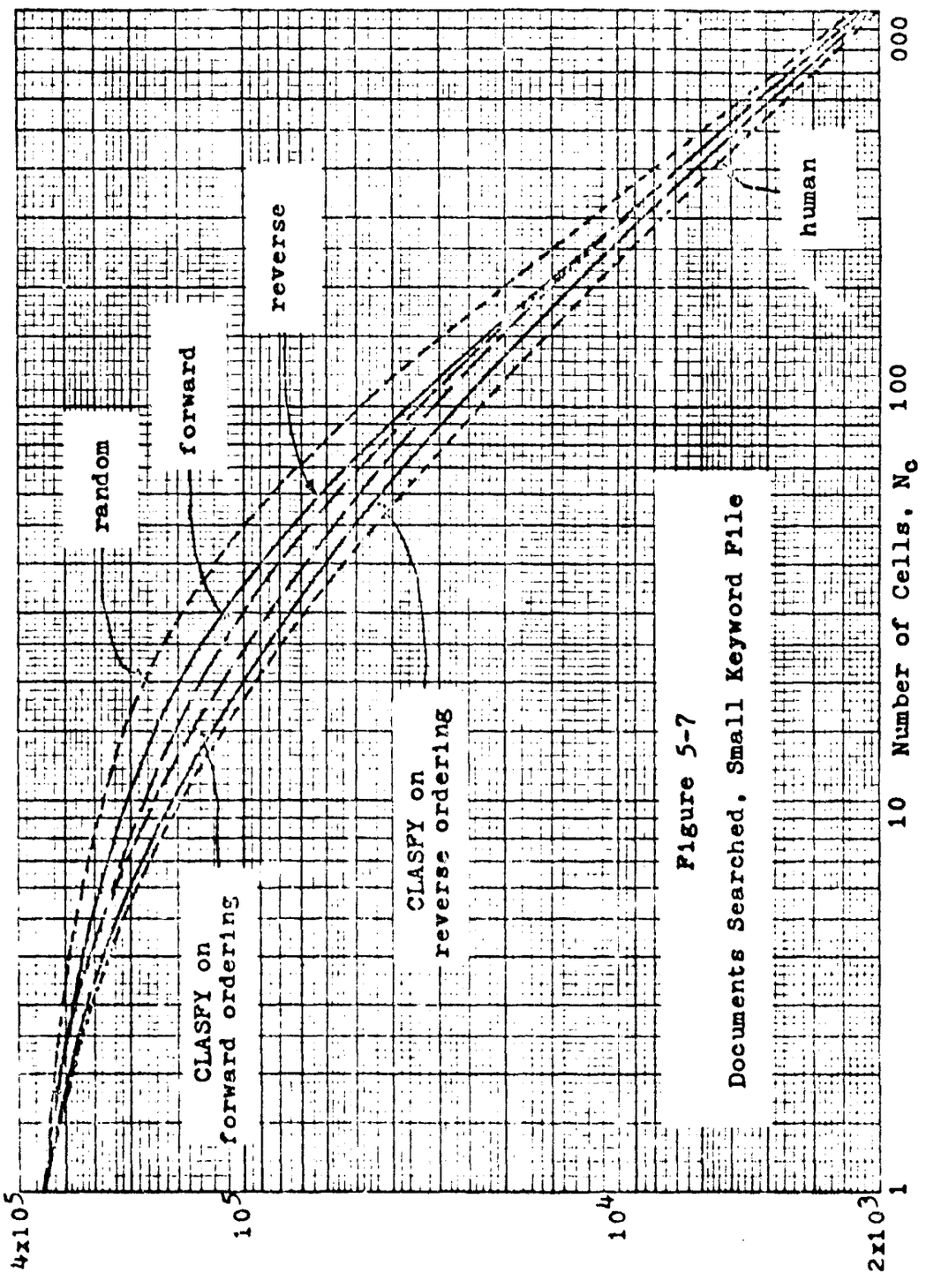

Documents Searched, 165 Requests
4x10^5
10^5
10^4
2x10^3
1
10
100
000
Number of Cells, N_c
random
forward
reverse
human
CLASPY on forward ordering
CLASPY on reverse ordering
Documents Searched, Small Keyword File

Figure 5-7

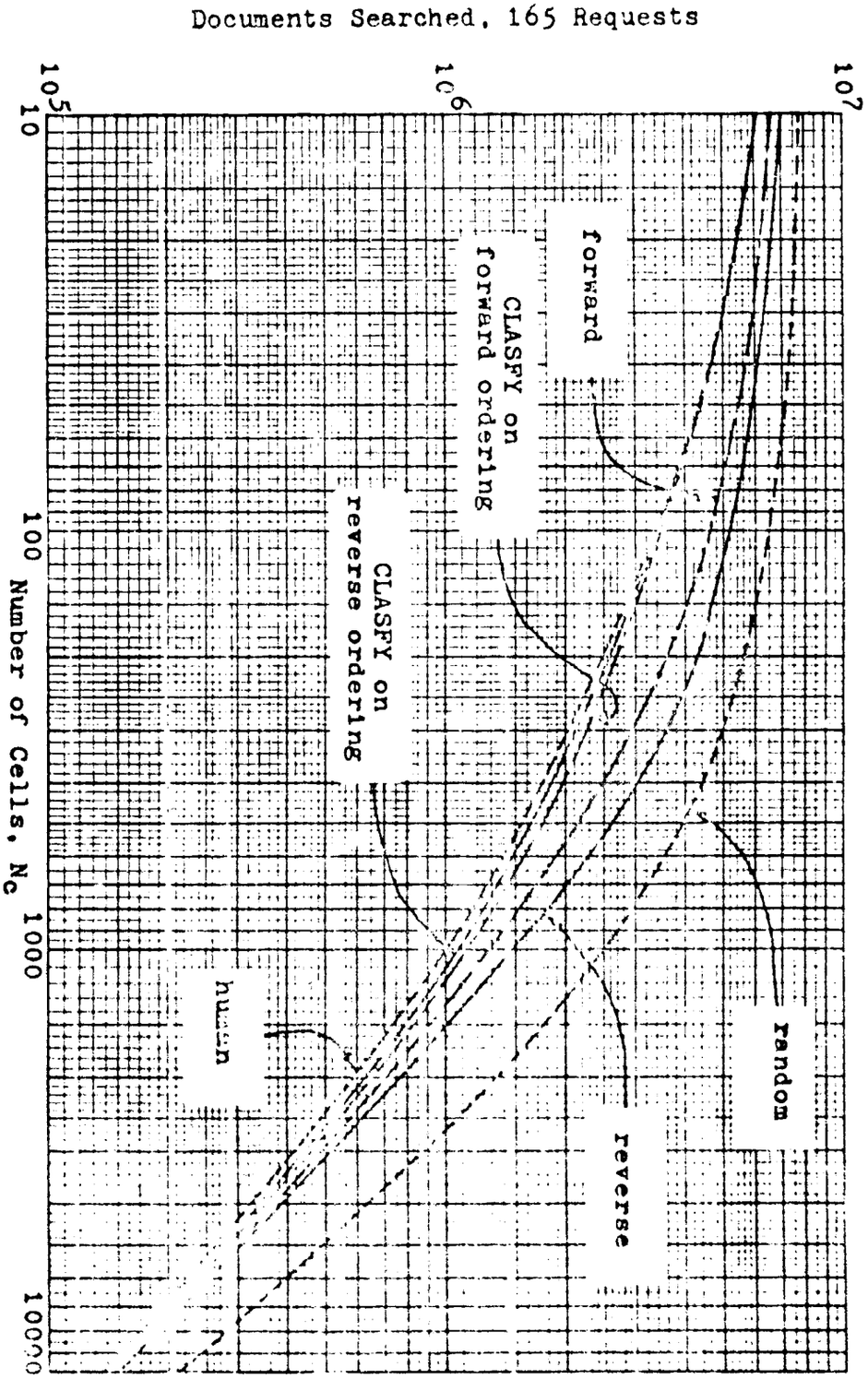


Figure 5.8

Documents Searched, Large Keyword File

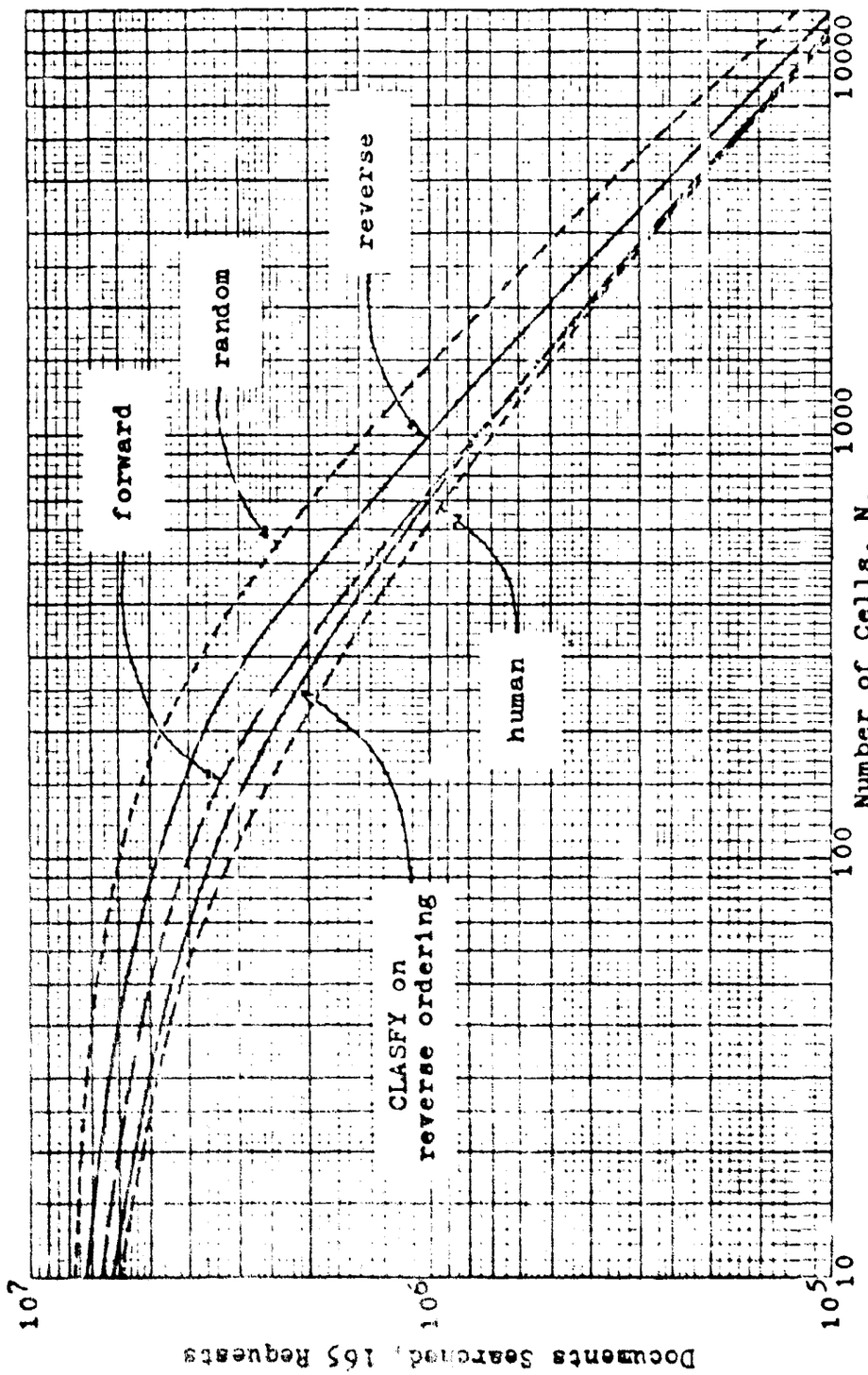


Figure 5-9

Documents Searched, Title Word File

the explanation of this phenomenon lies with the fact that the cells of the human classification are more even with respect to numbers of documents than those produced by CLASFY. Because a cell with more documents is more likely to be accessed than one with fewer documents, the size of the average cell searched in a system using CLASFY would be larger than that in a system using the human classification. Therefore, for the same number of cells searched, one would expect somewhat more documents searched with CLASFY than with the human system. The remainder of this effect (the above does not account for all of it) is attributed to the different characteristic of the classification systems.

The above effect can be seen in columns (a) and (b) of Table 5-2. It is noted that the percent documents searched is always greater than the percent cells searched. The number of cells shown represent the low end, logarithmic center, and high end of the range of interest.

Columns (c) and (d) of Table 5-2 show the percent and number of documents searched per document retrieved. Fortunately, the <u>percent</u> of documents searched per retrieval decreases (in the corresponding cell ranges of interest) with increasing file size. This results in the <u>number</u> of documents searched per document retrieved remaining essentially constant (see column (d), Table 5-2) with respect to file size. This is very significant, for if it holds for larger collections, it means that regardless of collection

File	Cells*	(a) % Cells Searched Per Request	(b) % Documents Searched Per Request	(c) % Documents Searched Per Document Retrieved	(d) Number of Documents Searched Per Document Retrieved
Small Keyword	30	18	20	3.9	87
	61	10	11	2.1	48
	125	4.9	5.8	1.1	25
Large Keyword	200	31	33	0.20	93
	548	18	20	0.12	57
	1500	9.0	10	0.062	29
Title Word	200	30	32	0.28	133
	548	14	16	0.14	66
	1500	6.2	6.9	0.060	28

* Low end, logarithmic center, and high end of ranges of interest.

Table 5-2

Percentage of Documents and Cells Searched, CLASFY

size, the same number of documents must be transmitted into main storage per document retrieved. Because the number of documents per cell would tend to be the same or greater for larger collections, this implies that the number of cells searched (and hence, number of memory accesses) per document retrieved would remain the same or be fewer for larger collections.

5.5 Quality of Hierarchy

For each file classification, up to a few hundred cells, done by CLASFY, a key-to-node table, a node-to-key table, and a terminal node table were produced (see Section 4.3). The cell limitation was imposed by the program used to produce the above tables.

5.5.1 Size of the Key-to-Node Table

The size of the key-to-node table (or equivalently, the node-to-key table) reflects, to some extent, the quality of a hierarchical classification system. For the same number of keys per cell, a smaller key-to-node table means that more keywords have migrated upwards in the tree, thereby producing a fuller hierarchy. In addition, a smaller key-to-node table means less storage space is required to store the table.

Because CLASFY is the only automatically generated hierarchical system being studied here, the hierarchies produced by CLASFY cannot be compared with those produced by other systems.

Figures 5-10, 5-11, and 5-12 show the size of the key-to-node tables for CLASFY with and without hierarchy generation. Of course, the size of a key-to-node table without a hierarchy (i.e., the only nodes are cells) is equal to the total number of keywords in the cells, or $N_{kc} \times N_c$. The curves for random and human classifications (no hierarchies) are shown for comparison.

From these figures, it can be seen that forming a hierarchy for a collection classified by CLASFY yields about a ten percent reduction in the size of the key-to-node table.

5.5.2 Subjective Evaluation and Example

The evaluation of the quality with respect to browsing of a hierarchy is necessarily subjective. The best way of doing this is to allow a number of users in the field of the collection to utilize the system (on-line) for a reasonable period of time and then present their opinions on the utility of the hierarchy for browsing purposes. However, since the retrieval system has not yet been implemented for on-line browsing, experiments of this type have not been performed.

Instead, subjective evaluations of the hierarchies obtained were made by the author and some associates. Attempts were made to extract a unifying title out of the keywords which appear at each node. After that was done,

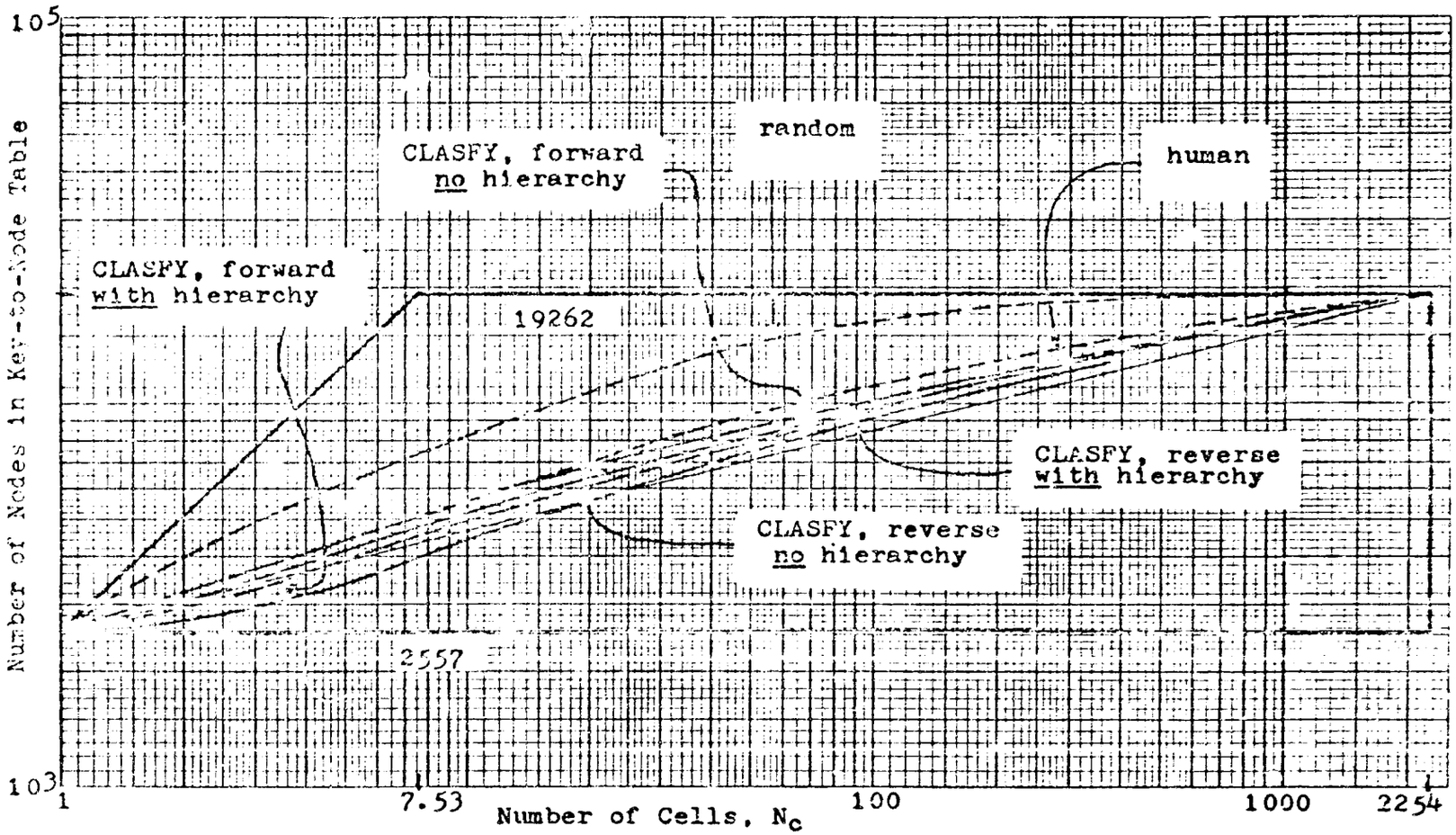


Figure 5-10

Size of Key-to-Node Table, Small Keyword File

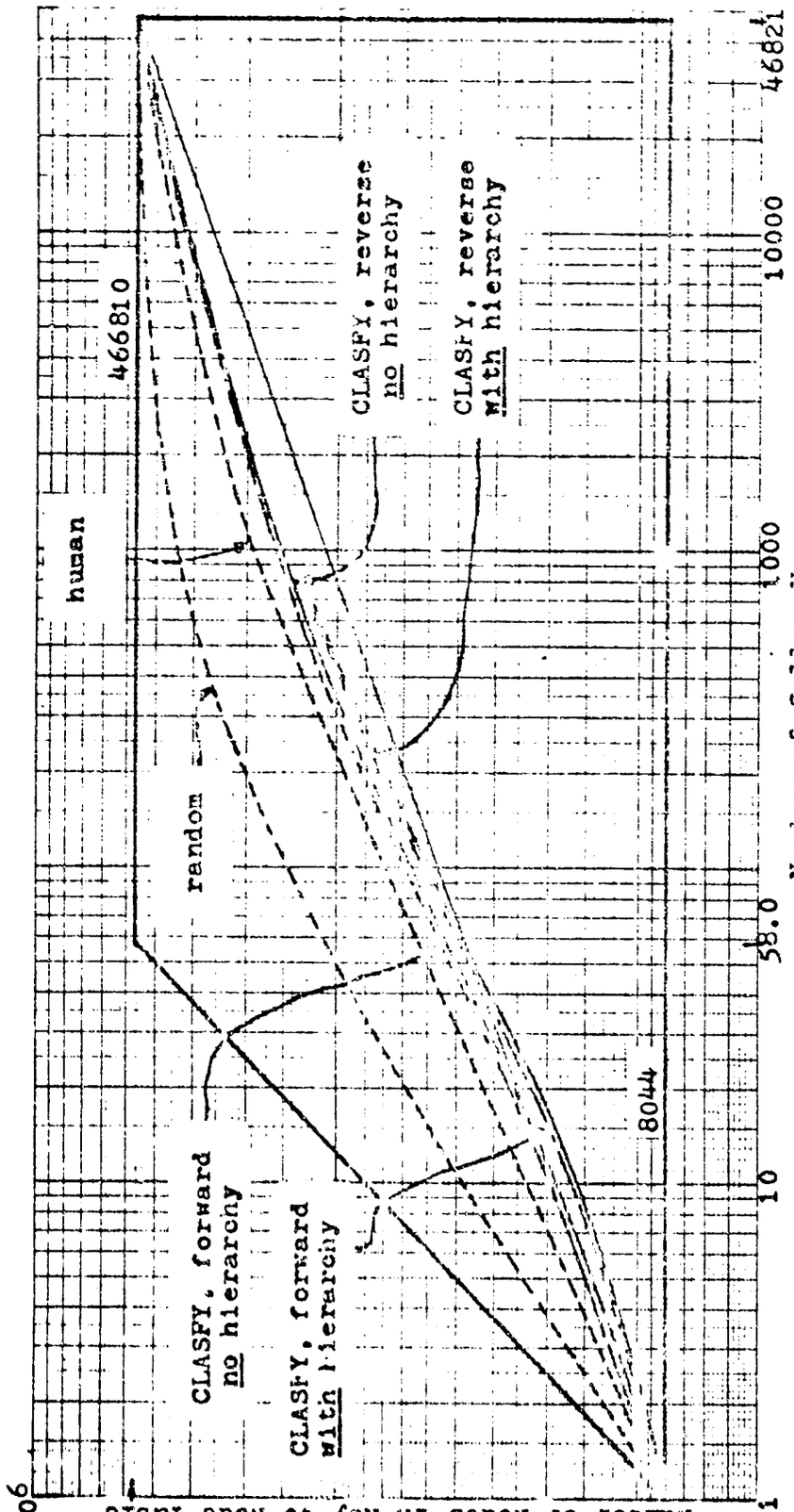


Figure 5-11

Size of Key-to-Node Table, Large Keyword File

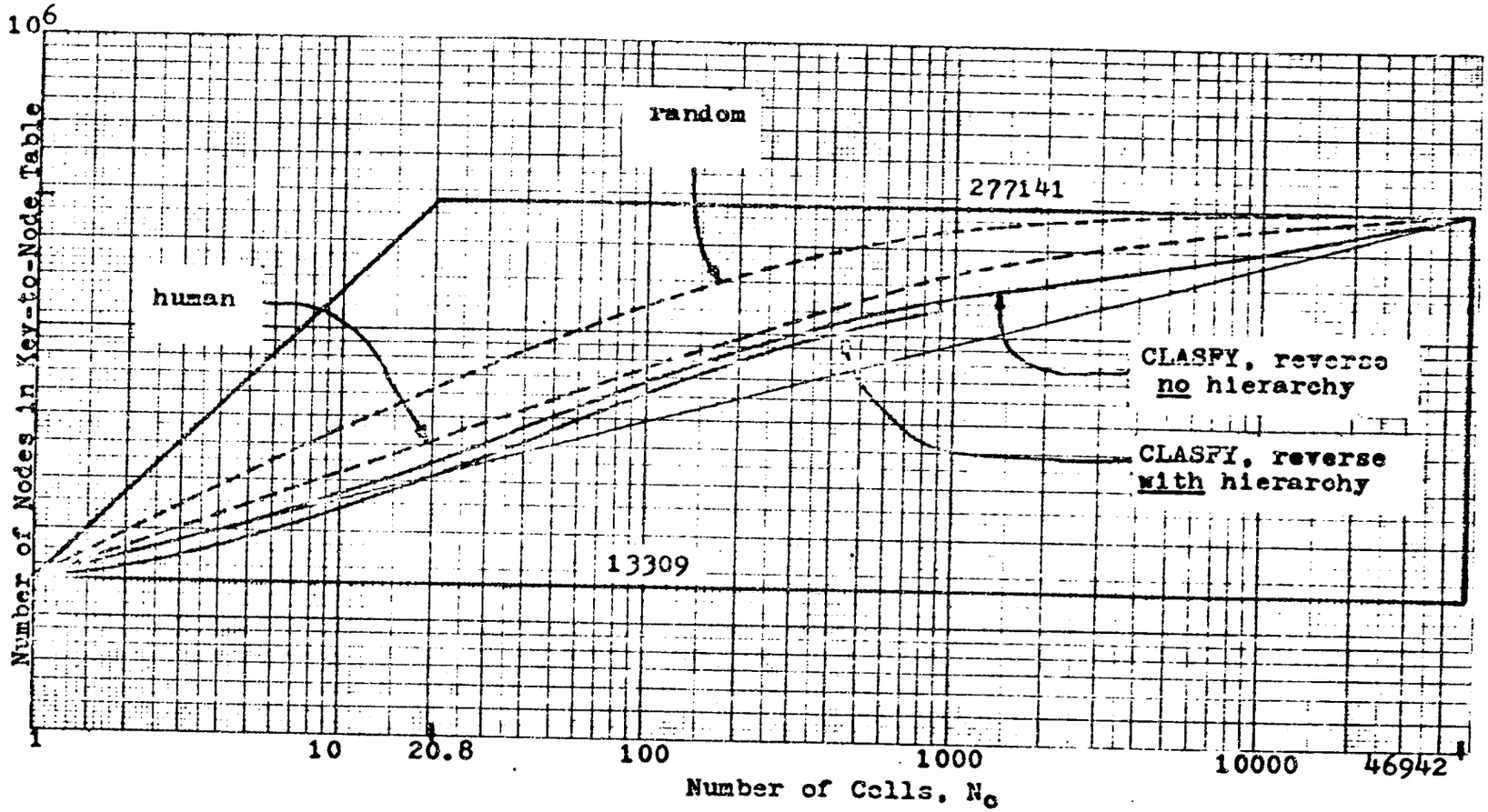


Figure 5-12

Size of Key-to-Node Table, Title Word File

each hierarchy was examined for its ability to separate subject areas and to see if the node titles indicate increasing specialization as one proceeds down a path in the tree.

This was not very successful for the samll keyword file. It was found that there were too few (2254) documents in the collection to provide reasonable subject separation at the various nodes in the hierarchy. On the other hand, the hierarchies generated from the output of CLASFY on the large files seem quite acceptable for $N_o > 100$.

The following example of portions of a hierarchy is taken from a classification of the large keyword file in reverse order by CLASFY. Appendix C presents the complete hierarchy for a similar classification performed by CLASFY. For this classification, the node stratification number, N, was set at 5, the sensitivity factor, E, was varied from 150 at the top of the hierarchy to 25 at the bottom, and the cell criterion, C, was set equal to a maximum of 460 documents per cell. As a result of the classification, 249 cells were produced. The average number of documents per cell is 46821/249 or 188 documents per cell. Including the apex and cells as levels, the resulting hierarchy tree varies from three to seven levels (if it were a balanced tree, it would have had four to five levels). The number of nodes at each level for the actual tree and a balanced tree is shown below. The numbers in parentheses represent the number of terminal nodes, or

cells produced at each level.

Level	Actual Tree Total Nodes	Cells	Balanced Tree Total Nodes	Cells
1	1		1	
2	5		5	
3	25	(4)	25	
4	105	(86)	125	(94)
5	95	(82)	155	(155)
6	65	(62)	-	
7	15	(15)	-	
Totals	311	(249)	311	(249)

Despite a few nodes with few or no keywords (e.g., nodes 1 and 1.2 have no keywords), in most cases it was not too difficult to summarize the keywords at a node. Figures 5-13, 5-14, and 5-15 show lists of some of the keywords at various nodes along with the node numbers and the manually formed titles for the nodes (the titles are just under the node numbers). The sample nodes were chosen to illustrate the hierarchical nature of a tree. For example (see Fig. 5-13), node 1.5.1, Organic Chamistry, is under node 1.5, Chemistry. Also node 1.2.3.2, Fission Products, is under node 1.2.3, Nuclear Explosions. All the nodes of Fig. 5-14 are in the same region of the tree (under but not necessarily directly under, node 1.1.1) and hence are relatively close in subject content. Figure 5-15 shows nodes at three levels of the bottom portion of part of the hierarchy. Nodes

Node 1.5

CHEMISTRY

chemical reactions
chemical analysis
reaction kinetics
absorption
stability
solutions
separation process
uranium
impurities
thermodynamics
decomposition
labelled compounds
thorium oxides
oxidation
electric potential
adsorption
lattices
cations
spectroscopy
polymers
salts
solubility
organic acids
chromatography
phenyl radicals
organic chlorine compounds
alcohols
phenols

Node 1.5.1

ORGANIC CHEMISTRY

organic compounds
organic nitrogen compounds
organic sulfur compounds
organic bromine compounds
organic fluorine compounds
methyl radicals
propyl radicals
isomers
amines
benzene
ethanol
ethers
urea
ammonia
acetic acid
nitric acid
heterocyclics
solvent extraction
polymerization
alkyl radicals
oxygen compounds
cycloalkanes
hydroxides
catalysis
amides
hexane

Node 1.2.3

NUCLEAR EXPLOSIONS

nuclear explosion
radioactivity
radioisotopes
contamination
environment
detection
gamma radiation
temperature
analysis
pressure
computers
radiation protection
safety
economics

Node 1.2.3.2

FISSION PRODUCTS

fission products
filters
decontamination
waste solutions
standards

Figure 5-13

Sample Nodes of Hierarchy, I

Node 1.1.1.1.2

RADIOTHERAPY AND MONITORING

man
performance
radiation protection
radiotherapy
safety
cancer
tumors
medicine
cobalt 60
shock waves
surgery
survival time
bibliography

energy
determination
scintillation counter
monitoring
dosemeters
standards
numericals
MEV range
operation

Node 1.1.1.3

ANALYSIS OF RADIATION EFFECTS ON ANIMALS

man
rats
mice
fish
age
body
bones
brain
RNA
drugs
kidneys
animals
rabbits
metabolism
animal cells
bone marrow
physiology
nervous system
leucocytes
blood cells
radiations
detection
tracer techniques
biochemistry
amino acids
strontium 85
antibiotics
toxicity
hormones
medicine
cesium 137
enzymes
calcium
proteins
recovery
carbon 14
genetics
cytology
phosphorus
membranes

Node 1.1.1.4

ENVIRONMENT AND METABOLISM

dogs
man
environment
metabolism
diet
food
meat
milk
proteins
plants
physiology
bones
growth
age
body
strontium 90
cobalt 60
performance
radiation protection
radioactivity
radioisotopes
fission products
detection
radium
radioautography
lungs
intestine
blood serum
rats
rabbits

Figure 5-14

Sample Nodes of Hierarchy, II

Node 1.2.2.2.1

PARTICLE MODELS

particle models
elementary particles
mass
scattering amplitude
vectors
protons
production

Node 1.2.2.2.1.1

THEORETICAL PHYSICS

quantum field theory
relativity theory
field theory
group theory
tensors
invariance principle
sum rules
parity
fermions
SU group
photons
decay
spin
cross sections

Node 1.2.2.2.1.2

STRONGLY INTERACTING PARTICLES

baryons
mesons
hyperons
quarks
isospin
matrices
pions
electric charges
elastic scattering
mathematics
angular distribution
energy
spectra

Node 1.2.2.2.1.2.2 (cell)

RESONANCES AND STRANGENESS

N* resonances
XI resonances
F resonances
K* resonances
Y* resonances
strangeness
strange particles
transients
decay
energy levels
phase shift
kaons-neutral
pions-plus
pions-minus
kaons
kaons-plus
hadrons
omega particles
omega-minus
antinucleons
hyperfine structure
tensors
bound state
Schrodinger equation
branching ratio
time-space
singularity
weak interaction
magnetic fields
MEV range
differential equations
magnetic moments
nuclear theory
monte carlo method

Node 1.2.2.2.1.2.3 (cell)

PARTICLE THEORIES

bootstrap model
current algebra
field theory
group theory
SU-2 group
SU-3 group
SU group
SU-12 group
O group
O-3 group
S-matrix
S-wave
strangeness
Legendre functions
Feynman diagram
conservation laws
hyperfine structure
Mossbauer effect
parity
inelastic scattering
coupling constants
Regge poles
dispersion relation
sum rules
statistics
transients
phase shift
G-parity
selection rules
Hamiltonian operator
pair production
integrals
orbits
annihilation
spin
equation
lattices
cross-sections
exitation
crystals
magnetism
neutrons
kaons
kaons-minus
pions-minus
fermions
bosons
antiprotons
phonons
antinucleons
omega particle
omega-minus
antihyperons
GEV range

Figure 5-15

Sample Nodes of Hierarchy III

1.2.2.2.1.2.2 and 1.2.2.2.1.2.3 are terminal nodes, or cells.

The entire hierarchy is too extensive to be shown here. Therefore, two sections of it have been selected and are displayed in Figures 5-16 and 5-17. For the sake of clarity, only the node numbers and manually generated node titles are shown. The tree segment of Figure 5-16 contains the nodes shown in Figure 5-14 and that of Figure 5-17 contains the nodes shown in Figure 5-15. In the tree segments, a dashed line represents the path to another node and a "c" represents the path to a cell. All the nodes of the tree segments shown are not labelled down to the cell level because (a) all the labels could not fit in the diagram and (b) all the nodes were not inspected for title assignment because of the large numbers of keywords and nodes (311) involved in this manual process.

The quality of a hierarchy should be measured by its convenience to the user, and not by how closely it matches an a priori, manually produced one. Therefore, this hierarchy will not be compared to the one of the human classification described in Section 4.6. The final evaluation of hierarchies such as this will have to await on-line tests of the type described in the beginning of this section.

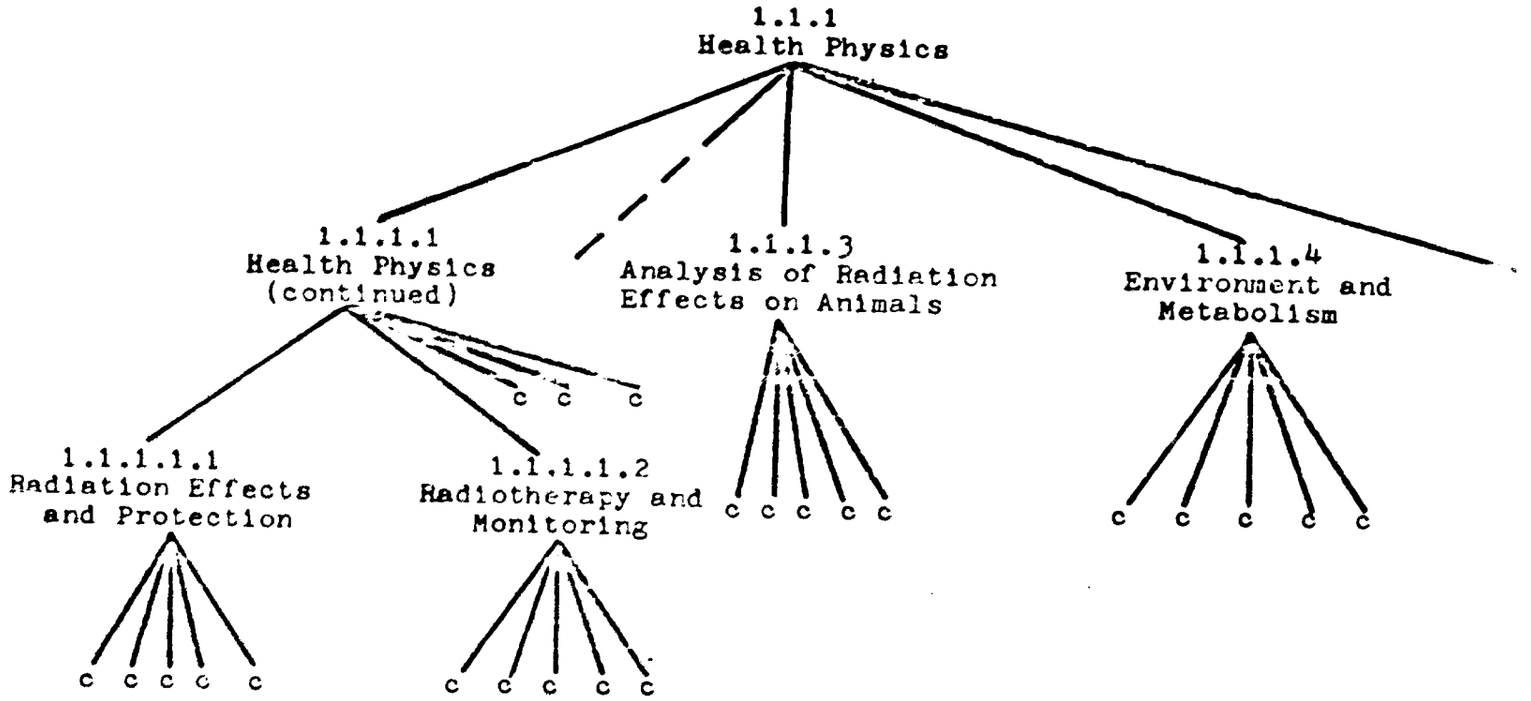


Figure 5-16

Portion of Hierarchy Tree, I

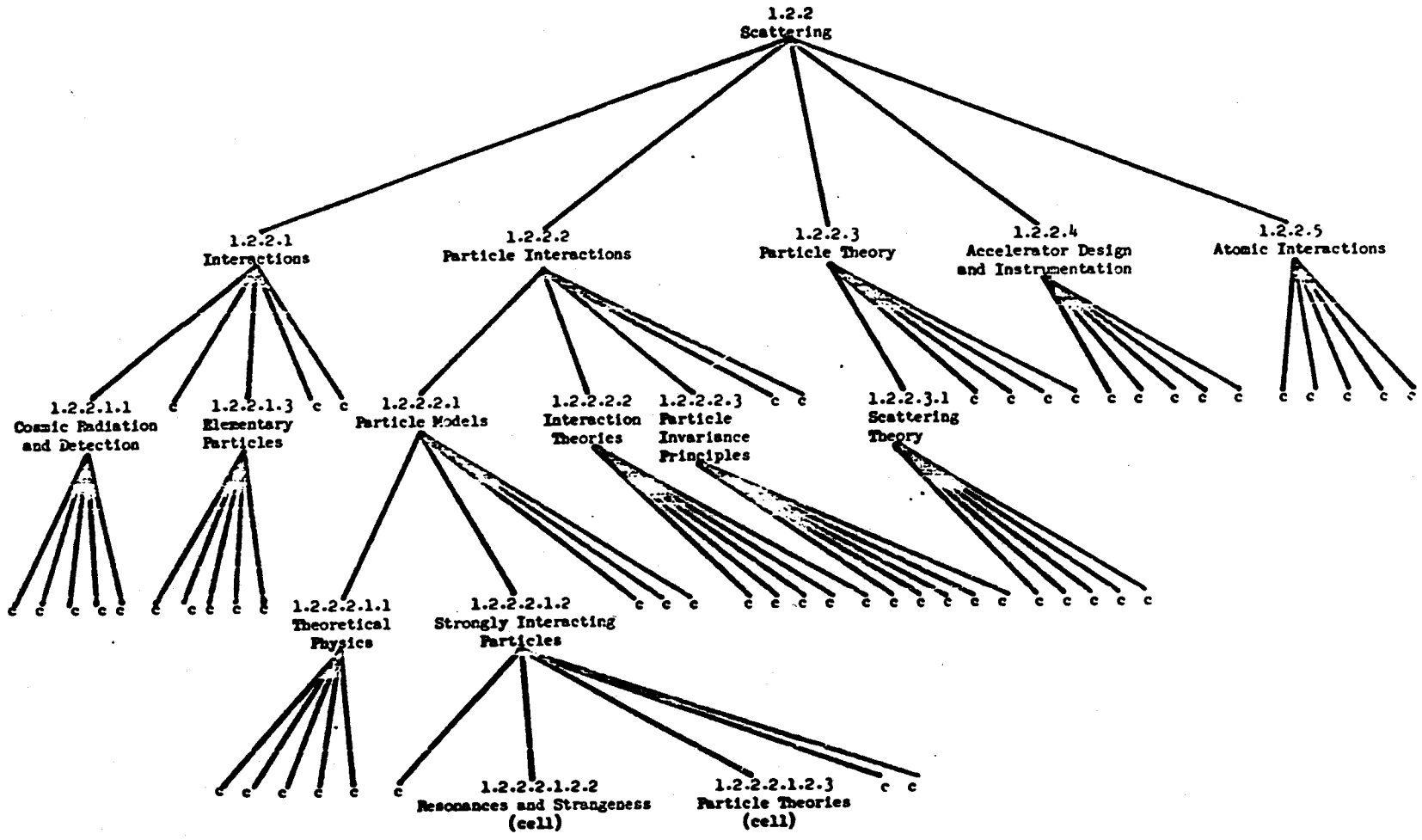


Figure 5-17

Portion of Hierarchy Tree, II

5.5.3 Documents in Cells

Most of the documents of each cell are very close in subject area. For example, consider the document descriptions shown in Figure 5-18. All of these documents were placed into terminal node 1.1.1.1.1.1, the leftmost cell shown under node 1.1.1.1.1, Radiation Effects and Protection, of Figure 5-16. The keyword numbers have been converted back into the English keywords. The documents shown in Figure 5-18 represent 10 out of 391 documents in that cell. The documents in the cell contain 2822 keyword occurrences, but only 216 distinct keywords (the ten documents of the example have 73 keyword occurrences and 20 distinct keywords).

It is interesting to compare this grouping of documents by CLASFY with that of the NSA classification system (from which the "human" system was derived). In Nuclear Science Abstracts, 7 out of the 10 documents were classified under "Radiation Effects on Plants". Documents 4789 and 34631 were classified under "Genetics and Cytogenetics" while 41382 can be found under "Ecology". It should be noted that documents 22785 and 34631, while placed in different NSA categories, agree in 7 out of 8 keywords and are both concerned with mutations of barley. In fact, document 41382, which was placed in a third category, also has many keywords in common with these two documents and discusses the same subject.

Document Number	Keywords			
1836	RADIATION EFFECTS MUTATIONS GENETICS	PLANTS AGRICULTURE	TESTING X RADIATION	NEUTRONS GAMMA RADIATION
4789	RADIATION EFFECTS MUTATIONS	PLANTS AGRICULTURE	USES	BIBLIOGRAPHY
16523	RADIATION EFFECTS CEREALS	MUTATIONS GENETICS	PLANTS BARLEY	ANALYSIS X RADIATION
22785	RADIATION EFFECTS CEREALS	MUTATIONS GENETICS	PLANTS BARLEY	TESTING X RADIATION
22792	RADIATION EFFECTS CEREALS	MUTATIONS GENETICS	BARLEY	X RADIATION
24673	RADIATION EFFECTS MUTATIONS GENETICS	PLANTS AGRICULTURE EXPANSION	USES TESTING	TIME IRRADIATION
26665	RADIATION EFFECTS CEREALS	MUTATIONS	BARLEY	RADIATIONS
34631	RADIATION EFFECTS CEREALS	MUTATIONS GENETICS	PLANTS BARLEY	TESTING
34818	RADIATION EFFECTS CEREALS	MUTATIONS BARLEY	X RADIATION	IRRADIATION
41382	CEREALS MUTATIONS	PLANTS BARLEY	X RADIATION USES	VARIATIONS PRODUCTION

Figure 5-18

Portion of Terminal Node 1.1.1.1.1.1

One should not conclude from the above that one set of categories for these documents is better than the other, but rather that there is more than one "reasonable" way of classifying documents and that a manual, a priori system does not necessarily categorize documents better than an automatic system.

Naturally, with 391 documents, the subject scope of this cell is much broader than that described by these 10 documents, and, in fact, contains documents on radiation effects on man as well as on plants. The two fields would probably be split apart if more than 249 cells were desired.

Unfortunately, automatically derived categories are prone to grossly misclassifying a number of documents or groups of documents. While manual systems are not exempt from errors, gross errors are rare. An example of this is a few documents on measurement and detection of cosmic radiation which were placed into the cell under discussion. Evidently they were placed into this cell via documents concerned with the effects of cosmic radiation. However, it is obvious that a better section of the hierarchy for these documents would be in a cell under node 1.2.2.1.1, Cosmic Radiation and Detection, shown in Figure 5-17. It is believed that the number of such misclassifications can be greatly reduced by modifying Pass 3 of the CLASFY algorithm to place documents with redundant descriptions

logically into a particular group instead of arbitrarily selecting a group as is sometimes done at present (see Section 4.2.1).

5.6 Summary of Results

The experiments described in this chapter led to consistent results which enables one to rank the quality (with respect to machine retrieval) of the classification systems studied.

As expected, all other systems outperformed the random classification. Next in increasing order of quality come the reverse and then the forward classifications. Unfortunately, these relatively simple classification schemes are not nearly as good as more sophisticated techniques, such as that embodied by CLASFY.

The results using CLASFY are uniformly good regardless of input ordering; however, CLASFY performs best with the input file in reverse order. It was found that the differences in results caused by the order (based on three orderings: forward, reverse, and random) in which the documents are presented to CLASFY decreases as the size of the collection increases. With respect to retrieval efficiency, CLASFY and the human, a priori system are very close in quality. CLASFY is slightly better in the number of cells searched, while the human classification is slightly better with respect to the number of documents searched.

This means that, because cell access time is usually longer than incremental document transmission and search time, if one system is to be ranked above the other, the edge would have to be given to CLASFY.

The hierarchies produced by CLASFY were not compared with those produced by any other system. However, subjectively they seem to be quite "reasonable" and could be very useful for on-line browsing. The placement of documents into particular cells (categories), while not always in agreement with manually derived document placements (agreement is not necessarily desirable), in most, but not all, cases is quite satisfactory.

For comparison purposes, the entire hierarchy of a classification similar to the one discussed here is included as Appendix C of this dissertation.

5.7 Bonus Result - Average Length of Search in Serial Files

A question relating to serial files has come to the attention of the author and while not directly related to automatic classification, can easily be answered as a result of some of the experiments performed here.

Fossum and Kaskey [48], among others [36,145], have proposed organizing serial files via some form of keyword ordering in order to avoid having to search all the documents in the file for each request. For example, consider the following documents ordered by increasing keyword numbers:

Document	Keywords
C	1 2 3 4
F	1 2 5
A	1 4 7
B	2 3 6
D	2 5 7
E	3 6

If a request of keywords 1 AND 3 is entered, the serial search may stop after three documents (i.e., after document A) for one is guaranteed that "1" does not appear further on in the file.

Possum and Kaskey [48] state:

> "Does this approach have any significant potential in a document retrieval application? Unquestionably, it permits terminating a search without examining all the documents in the file and, from this standpoint, is preferable to a straight document-sequenced organization. The percentage of the file records that can be bypassed, on the average, has not been reported. In fact, so far as known, the proposal has not been tested against an actual file of document descriptions and a representative sample of search requests."

In the experiments reported here, such a file of document descriptions and sample of search requests were available. The forward and reverse orderings (before modification) are, in fact, orderings of the file based on keyword numbers.

The 165 search requests were applied to the forward and reverse orderings of both large files. The per-

centage of the files serially searched per request is as follows:

File	Order	Percentage Documents Searched per Retrieval Request
Large Keyword	Forward	95.4
	Reverse	84.0
Title Word	Forward	91.4
	Reverse	85.5

The numbers only apply to individual requests as in an on-line system. Batching of requests would eliminate any advantages to be gained by file ordering.

No attempt was made to optimize the file ordering or the keyword numbering in order to minimize the number of documents searched. However, based on the above results, it would seem that one would not be able to reduce the percentage of documents searched below about 80% by keyword and document ordering. This reduction is not enough to allow the use of serial files in large-scale on-line IS&R systems. However, small on-line systems, where there are few enough documents to allow for serial searching, might be able to profit from the above file organization.

CHAPTER 6

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

6.1. General Conclusions

On the basis of the experiments on almost 50,000 documents described in Chapter 5, it is concluded that automatic classification can go a long way towards solving some of the problems of large-scale information storage and retrieval.

An *a posteriori* automatic classification system (CLASFY) has been described which was shown, by a number of different measures, to be *at least* equal in classification quality to a manual, *a priori* classification system. However, because of its automatic and flexible nature, it is felt that automatic classification can be vastly superior to any manual system. This statement is made taking into account the realization that CLASFY, while a perfectly respectable system, can stand some improvements and is probably far from the ultimate (if there is such a thing) in automatic classification systems.

Regardless of the quality of classification, an automatic classification system will only be used if the classification time required for large files is reasonable. It was found (see Section 5.3) that CLASFY took about 1¼ hours per tree level to classify about 50,000 document descriptions on an IBM 7040 computer. It is expected that

this time could be reduced by at least an order of magnitude by the use of a modern, high-speed computer (it could be further reduced by multi-processing - a technique to which CLASFY lends itself as a result of the independence of processing at each node). This is because of the relatively slow processing speed (basic cycle time = 8 microseconds, add time = 16 microseconds) of the 7040 and, of even greater importance, the relatively slow and limited secondary storage facilities available (at least one fourth of the time was spent in just copying data from disk to tape and rewinding tapes, processes which would not have to be done if more disk were available). In addition, the 7040 used for these experiments was being operated on-line, i.e., all processing stopped during printing of the node summaries (see Figure 4-2).

Table 6-1 presents the approximate classification times required by CLASFY to operate on the files of the examples of Tables 1-1, 1-2, and 2-1. As seen in Table 6-1, the time required to classify 10^6 and 10^7 documents should be no more than 12 and 120 hours, respectively. This is quite reasonable, especially considering that the number of books in the Library of Congress is about 10^7. These times represent .043 seconds per document.

Number of documents, N_d	10^6	10^7
Number of cells, N_c	10^4	5 · 10^4
Number of documents per cell, N_{dc}	100	200
Stratification number, N for $\log_N N_d \approx 7$ (see last paragraph of Section 4.2.4.1)	7	10
Average number of classification levels = number of levels in tree minus one	4.7	4.7
Approximate 7040 time per level	25 hrs.	250 hrs.
Approximate total 7040 time required	120 hrs.	1200 hrs.
Maximum time required using modern, high-speed computers	12 hrs.	120 hrs.

Table 6-1

Classification Time for 10^6 to 10^7 Documents using CLASFY

6.2 Future Research

There are a number of directions to further research in the area of automatic classification, some obvious and some not so obvious.

One obvious direction is to improve the CLASFY algorithm. Some means for accomplishing this were mentioned or implied in the text of this paper. Another direction of research, also slanted towards CLASFY, is to establish an on-line IS&R system using CLASFY for automatic classification. This would enable one to obtain user reactions, particularly with respect to the quality and utility of browsing.

Other classification systems should be designed (some already exist) and applied towards the classification of large files such as the one used in these experiments and then compared on the basis of the measures described in Section 5.2. If this could be done with reasonable uniformity, an IS&R system designer will have a basis upon which to select one classification system over another, something which is lacking at present.

Another area of research is that of retrieval statistics. It is desirable to have some idea of how many documents will be retrieved before the actual retrieval takes place. A user might modify a request based upon this number. For example, if 2000 documents are estimated for retrieval, a user would probably want to narrow the re-

quest before the actual retrieval takes place. In an inverted file, one can obtain complete statistics at the expense of manipulating long lists of document numbers. In a serial file, on the other hand, few, if any, statistics are available.

The retrieval statistics for a file organized on cells are somewhat in between those of the serial and inverted files. One knows how many cells will be accessed and how many documents are in those cells, but not how many documents in those cells will satisfy the search request. This can be estimated a number of different ways, some of which are:

1) On the basis of the average number of documents searched per retrieval (see Table 5-2 of Section 5.4.3).
2) On the basis of the number of documents retrieved from searching a sampling (maybe ten percent) of the cells to be searched.
3) On the basis of the number of request keywords found in the cells which satisfy the request.

The search for the method which achieves the best retrieval statistics is an interesting subject for future investigation.

APPENDIX A

NUCLEAR SCIENCE ABSTRACTS DATA FILES

A.1 Source of the Data

The data used in these experiments was obtained from the Atomic Energy Commission, Division of Technical Information Extension through the aid of Joel O'Connor, formerly Chief, Computer Operations Branch of the above division. The data comprises parts of two out of three sets of data made available by the AEC upon request by qualifying research projects. These files are in machine-readable form on magnetic tape.

These files provide data on each document abstracted in Nuclear Science Abstracts (NSA). It should be noted that the three files contain different aspects of the same documents. The three files consist of:

1) Keyword File. Document identification plus descriptors (called "selectors" by the AEC) manually indexed from the EURATOM Thesaurus [46,47]. Discussed in detail in Section A.2.
2) Entry File. Document identification plus bibliographic data such as title, a priori category, language, journal citation, contract number, etc. Discussed in detail in Section A.3.
3) Subject Heading File. Document identification

plus subject headings used to index documents in NSA. There are about 33,000 headings [141] with an average of about four being used to index each document. This file was not used in these experiments and therefore will not be described in any further detail.

The actual files obtained for this research (on seven magnetic tapes) are the Keyword and Entry (titles) files of NSA Volume 22, Number 3 (February 15, 1968) - 2258 documents and Volume 21 (1967) - 47,055 documents. The Entry file of Vol. 22, No. 3 was not used in the experiments.

A.2 Keyword Files

Each document covered in NSA is assigned EURATOM indexing terms by subject specialists. This is done in order to add those documents covered by NSA to the collection of the Center for Information and Documentation (CID) of the European Atomic Energy Community (EURATOM). As of September 1966, that collection held 360,000 documents and was growing at the rate of about 120,000 documents per year. Rolling [111] presents a description of the EURATOM-CID system (batched searches on a serial file are performed).

The EURATOM Thesaurus [46,47] is similar to other current thesauri [58,92,137] except that the usual forms

of cross referencing (i.e., related broader, and narrower terms) are presented graphically. The thesaurus contains 15,695 usable terms plus 3,488 "forbidden" terms. However, in indexing for NSA, a number of additional terms were used where appropriate. If deemed desirable, some of these terms will be incorporated in future editions of the Thesaurus. As of early 1968, 15,517 different index terms had been assigned to NSA documents.

The keyword file[142] contains each document's abstract number, type, assigned category (called "Section Subsection code), and list of keywords. The abstract number is used to identify the documents. The types of documents indexed are: books, theses, conference papers, engineering materials letters, journal literature, patents, reports, and translations. The type information was not used in these experiments. The NSA categories are an *a priori* classification of the knowledge covered by the documents abstracted in NSA. There are almost 300 categories. This is the *a priori* classification referred to in Chapters 4 and 5 (see Section 4.6 for more details).

In addition to the actual English keywords and the NSA codes assigned to them, the keyword list also includes "link" (called "split" in ASC literature) information. The function of links is to group keywords such that the keywords in a group represent topics covered in the paper. By eliminating retrieval on conjunctions

involving keywords in different links, one reduces the number of "false drops" in response to a query. The wisdom of using links (and "roles") is the topic of numerous papers [32,66,86,123,134,144], some for and some against. The position taken here is that the utility of links is probably reduced in a hierarchical, on-line system. Hence, all link information was deleted in processing the keyword files.

Due to missing data and bad tape records, not all the documents from each file could be processed. In addition, for reasons of speed and economy of storage, documents with more than 47 keywords were deleted. Since this represented only 0.18% and 0.39% of the small and large files respectively, this should have little effect on the experiments. The actual number of documents used were 2254 (out of 2258) and 46,821 (out of 47,055).

Figure A-1 presents a macro-flowchart of the procedure used to prepare the keyword files for classification experiments. All file processing was performed on an IBM 360/65 using PL/I. The result of this process was to replace the English keywords with rank numbers which corresponded to the frequency of the English word. For example, the keyword "reactors" was replaced by its order number, "1". Figure A-2 shows the 200 highest occurring (large file) keywords and their frequencies of occurrence. Now, not only can the new keywords (i.e., numbers) be

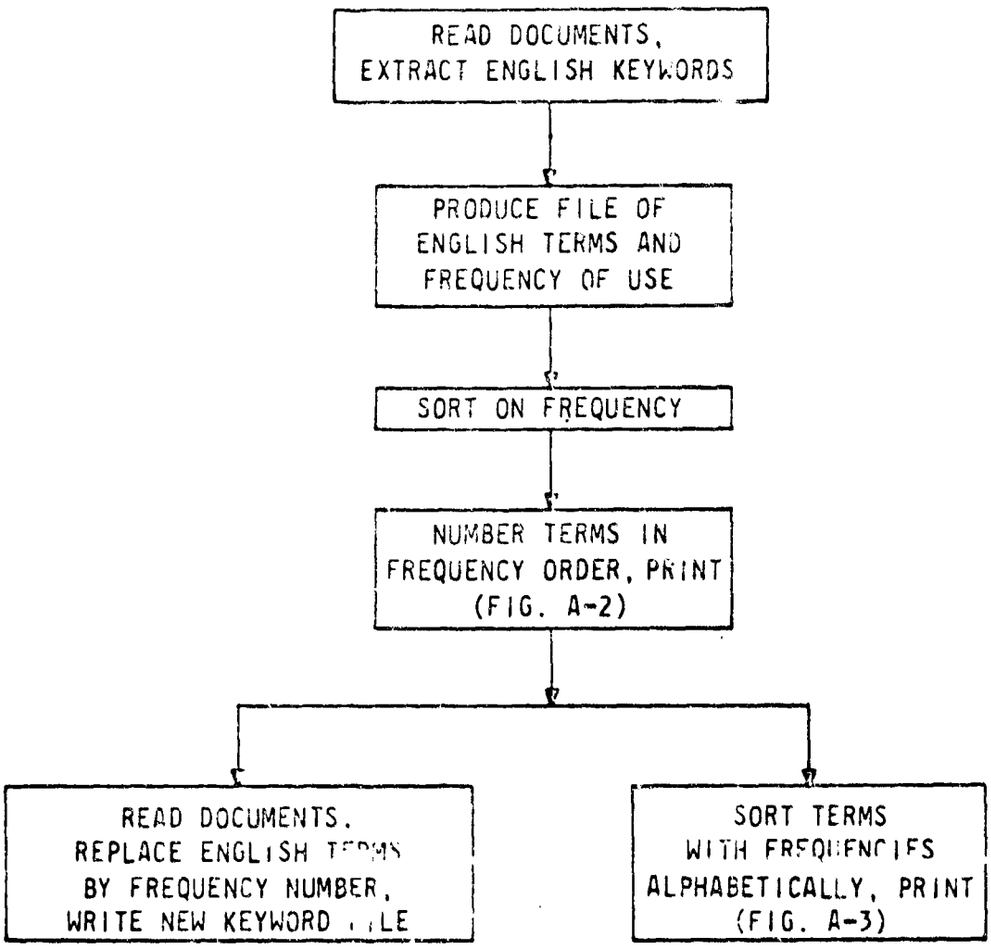


FIGURE A-1

MACRO-FLOWCHART OF KEYWORD FILE PREPARATION

NUMB	FREQ	TERM	NUMB	FREQ	TERM	NUMB	FREQ	TERM	NUMB	FREQ	TERM
1	4294	REACTORS	51	1355	NEV RANGE	101	849	CURRENTS	151	647	ASTROPHYSICS
2	4292	DESIGN	52	13-7	PIONS	102	827	SUPERCONDUCTIVITY	152	644	URANIUM DIOXIDE
3	4197	MEASUREMENT	53	13 5	NUCLEONS	103	826	EXPANSION	153	642	ULTRAVIOLET RADIAT
4	4065	RADIATION EFFECTS	54	13 3	FUEL ELEMENTS	104	825	PREPARATION	154	636	COSMIC RADIATION
5	4013	INTERACTIONS	55	13.7	STABILITY	105	819	MESONS	155	635	FUSES
6	3754	GAMMA RADIATION	56	1296	DIFFERENTIAL EQUAT	106	807	BERYLLIOSIS	156	634	TRACER TECHNIQUES
7	3544	ELECTRONS	57	1294	DIFFUSION	107	801	ECONOMICS	157	631	ANIMALS
8	3341	SPECTRA	58	1290	GASES	108	798	RESEARCH REACTORS	158	631	MATRICES
9	3243	TEMPERATURE	59	1282	POWER PLANTS	109	797	FISSION PRODUCTS	159	627	TRANSIENTS
10	3058	RADIATION DOSES	60	1260	NUCLEI	110	783	NUCLEAR MODELS	160	625	FLUID FLOW
11	3013	SCATTERING	61	1211	RADIATIONS	111	781	NEUTRON BEAMS	161	624	CANCER
12	2926	IRRADIATION	62	11 4	ORGAN	112	773	BEAMS	162	624	REACTIVITY
13	2844	NEUTRONS	63	1143	COMPUTERS	113	760	RADIATION CHEMISTR	163	622	POWER
14	2745	X RADIATION	64	1142	FIELD THEORY	114	756	PHOTONS	164	622	REACTOR CORE
15	2718	PLASMA	65	1140	TISSUES	115	755	IONIZATION	165	618	STANDARDS
16	2722	TIME	66	1137	REACTION KINETICS	116	753	HELIUM	166	615	STRESSES
17	2691	PRODUCTION	67	1132	RADIATION PROTECTI	117	753	METABOLISM	167	613	DENSITY
18	2532	CROSS SECTIONS	68	1083	CRYSTALS	118	745	LIQUIDS	168	600	KAONS
19	2507	VELOCITIES	69	1079	EMISSION	119	734	DEFECTS	169	594	CARBON
20	2494	PROTONS	70	1064	OXYGEN	120	729	RELATIVITY THEORY	170	592	METALS
21	2383	MAGNETIC FIELDS	71	1057	OPERATION	121	723	SAFETY	171	591	TUMORS
22	2360	ENERGY LEVELS	72	1037	SOLUTIONS	122	722	ATMOSPHERE	172	590	NITROGEN
23	2269	ENERGY	73	995	MASS	123	721	ALUMINUM	173	589	ACCELERATORS
24	2231	ABUNDANCE	74	986	SEPARATION PROCESS	124	717	ELECTRON BEAMS	174	588	COPPER
25	2145	ANALYSIS	75	984	RADIOACTIVITY	125	716	LOW TEMPERATURE	175	587	DECOMPOSITION
26	2073	NUCLEAR REACTIONS	76	981	URANIUM	126	715	ANIMAL CELLS	176	587	IRON
27	2019	TESTING	77	959	EXCITATION	127	714	CHARGED PARTICLES	177	586	LABELED COMPOUNDS
28	1959	WATER	78	957	FUELS	128	713	ALPHA PARTICLES	178	586	SODIUM
29	1936	USES	79	926	FAST NEUTRONS	129	711	THERMAL NEUTRONS	179	583	SHIELDING
30	1907	DISTRIBUTION	80	910	NEUTRON FLUX	130	700	ELASTIC SCATTERING	180	577	RADIOSENSITIVITY
31	1878	RESONANCE	81	956	CHEMICAL REACTIONS	131	698	ERRORS	181	574	SEMICONDUCTORS
32	1875	DETERMINATION	82	952	FREQUENCY	132	693	HEAT TRANSFER	182	573	DEUTERIUM
33	1766	MAN	83	939	SUN	133	686	INSTRUMENTS	183	567	OXIDATION
34	1742	CONFIGURATION	84	938	RADIOISOTOPES	134	684	HEATING	184	562	AIR
35	1734	RADIATION INJURIES	85	929	SURFACES	135	683	DETECTION	185	562	ARGON
36	1681	QUANTUM MECHANICS	86	923	ATOMS	136	683	MOLECULES	186	562	CALCIUM
37	1593	PERFORMANCE	87	913	RADIATION DETECTOR	137	682	PLANTS	187	560	BIBLIOGRAPHY
38	1562	EFFICIENCY	88	912	IMPURITIES	138	681	DEV RANGE	188	558	PARITY
39	1524	MATHEMATICS	89	911	GROUP THEORY	139	681	RADIO WAVES	189	558	PROTON BEAMS
40	1517	IONS	90	911	STARS	140	681	SCATTERING AMPLITU	190	547	MIXING
41	1513	SKIN	91	908	MOTION	141	680	SCINTILLATION COUN	191	547	NUCLEAR EXPLOSIONS
42	1488	NUMERICALS	92	907	VELOCITY	142	679	IONOSPHERE	192	546	BETA PARTICLES
43	1484	EQUATIONS	93	894	FABRICATION	143	677	EARTH	193	542	METHYL RADICALS
44	1472	PRESSURE	94	893	PARTICLE MODELS	144	676	ORGANIC NITROGEN C	194	541	VECTORS
45	1466	LATTICES	95	893	RADIOTHERAPY	145	674	ELECTRIC CONDUCTIV	195	536	FALLOUT
46	1429	ABSORPTION	96	890	RATS	146	668	OSCILLATIONS	196	534	SENSITIVITY
47	1420	ELEMENTARY PARTICL	97	887	MOMENTUM	147	667	PULSES	197	534	STRONTIUM 90
48	1370	ANGULAR DISTRIBUTI	96	887	PROGRAMMING	148	664	THERMODYNAMICS	198	533	MAGNETOHYDRODYNAMI
49	1370	HYDROGEN	99	881	ELECTRIC CHARGES	149	663	STATISTICS	199	530	DEFORMATION
50	1366	HIGH TEMPERATURE	100	880	POLARIZATION	150	651	MICE	200	527	HETEROCYCLICS

Figure A-2

200 Most Frequently Occurring Keywords - 46821 Documents

easily manipulated by computer, but keywords can be compared on the basis of frequency by inspection. Figure A-3 presents a sample of the alphabetic listing of English keywords along with their corresponding frequencies and keyword numbers. Identical, but independent, processing was done for the small keyword file.

Statistics for these files can be found in Section A.4.

A.3 Entry (title word) File

A.3.1 Nature of the File

The NSA Entry File[140,142] contains each document's abstract number, type, assigned category, title, and other bibliographic information depending upon the type of document. The type and category are the same as described in the previous section. A semi-automatic procedure was used to obtain index terms from the document titles. However, at times, the above bibliographic material included a "short title". A short title is composed by an evaluator (not the author) when the original full title is not suitable. This can occur under various circumstances, such as a foreign language title, a title which includes a subtitle, lengthly titles, and cases where uniform abbreviation of words is desirable. However, since the short title usually did not change the significant words of the title but did simplify processing, the short title was used

NUMB	FREQ	TERM	NUMB	FREQ	TERM	NUMB	FREQ	TERM	NUMB	FREQ	TERM
7440	1	RADIOACTIVE MATERI	1602	32	RADON 220	221	484	REACTOR SAFETY	4201	4	RESERVOIRS
75	984	RADIOACTIVITY	1898	22	RADON 222	4198	4	REACTOR SIMULATORS	1183	58	RESIDUES
2930	9	RADIOAPPLICATORS	7458	1	RAFFINATES	2932	9	REACTOR SITES	751	123	RESINS
690	140	RADIOAUTOGRAPHY	7459	1	RAFFLE-CODE	3813	5	REACTOR SITING	1462	39	RESISTORS
640	105	RADIOBIOLOGY	4747	3	RAG	1	4294	REACTORS	[illegible]	234	RESOLUTION
597	173	RADIOCHEMISTRY	7460	1	RAILWAY	2794	10	REAGENTS	7486	1	RESOLUTION POWER
5583	2	RADIOCOLLOIDS	675	139	RAIN	7474	1	REAX-CODE	31	1878	RESONANCE
4192	4	RADIODERMATITIS	5587	2	RAJASTHAN POWER RE	452	243	RECOILS	1579	29	RESONANCE ESCAPE P
990	80	RADIODIAGNOSIS	1344	21	RAMAN EFFECT	691	140	RECORDING SYSTEMS	1077	70	RESONANCE INTEGRAL
3810	5	RADIOFREQUENCY	7461	1	RAMJET	360	298	RECOVERY	2038	19	RESONANCE NEUTRONS
327	329	RADIOGRAPHY	7462	1	RAMSAUER EFFECT	1019	76	RECRYSTALLIZATION	4751	3	RESONANCE PARAMETE
5504	2	RADIOIMMUNOASSAY	7463	1	RANA	7475	1	RECTIFICATION	3057	8	RESORCINOL
64	933	RADIOISOTOPES	2076	18	RANDOM PHASE APPRO	3814	5	RECTIFIERS	7487	1	RESPIRATION
241	448	RADIOLYSIS	5588	2	RANDOMNESS	2658	11	RECTUM	3817	5	RESPIRATORY EQUIPM
2359	14	RADIOMETERS	1574	33	RANKINE CYCLE	7476	1	[illegible] PROCESS	[illegible]	8	RESPIRATORY TRACT
7441	1	RADIOMETRY	7464	1	RANKINE-HUGONIOT F	3815	5	RECYCLING	1899	22	RESPONSE
1943	21	[illegible]	7465	1	[illegible]	7477	1	[illegible]	7488	1	[illegible]
4193	4	RADIOPROTECTIVE SU	7466	1	RAPID-CODE	3520	6	REDOX PROCESS	[illegible]	2	[illegible]-CODE
[illegible]	577	RADIOSENSITIVITY	2277	15	[illegible]	7478	1	REDUCTASE	2145	17	RETENTION
2931	9	RADIOSTERILIZATION	359	297	RARE EARTHS	495	220	REDUCTION	5597	2	RETICULAR CELLS
3095	8	[illegible]	1156	61	RARE GASES	1296	49	REFINING	2796	10	RETICULOCYTES
95	893	[illegible]	3312	5	RAREFIED GASES	469	235	REFLECTION	1741	27	RETICULOENDOTHELIA
4745	3	RADIOTOXINS	4196	4	RARITA-SCHWINGER E	769	119	REFLECTORS	1655	30	RETINA
3611	5	RADISHES	1603	32	RATE METERS	3816	5	REFLEXES	3818	5	RETINOBLASTOMAS
606	170	RADIUM	96	890	RATS	1249	52	REFRACTION	5598	2	REVIEW
7442	1	RADIUM BROMIDES	7467	1	RATS-CODE	830	107	REFRACTORIES	7489	1	REXALLOY 33
7443	1	RADIUM CHLORIDES	7468	1	RAVEN CODE	7479	1	REFRIGERATORS	1557	30	REYNOLDS NUMBER
7444	1	RADIUM COMPOUNDS	5589	2	RAYLEIGH SCATTERIN	862	102	REGENERATION	5599	2	RFT
4746	3	RADIUM ISOTOPES	7469	1	RAYLEIGH WAVES	414	267	REGGE POLES	7490	1	RG-1
7445	1	RADIUM NITRATES	7470	1	RAYLEIGH-SCHROEDIN	7480	1	REGULATIONS	891	98	RHENIUM
7446	1	RADIUM 219	4748	3	RAYLEIGH-TAYLOR IN	7481	1	REINDEER	[illegible]	113	RHENIUM ALLOYS
5585	2	RADIUM 220	3519	6	RB-1	120	729	RELATIVITY THEORY	7491	1	RHENIUM BORIDES
4194	4	RADIUM 223	7471	1	RB-2	1209	56	RELAXATION	5630	2	RHENIUM BROMIDES
2450	13	RADIUM 224	1208	55	RBE	2933	9	RELAXATION LENGTH	[illegible]	2	RHENIUM CARBIDES
7447	1	RADIUM 225	5590	2	RBU-CODE	2795	10	RELAYS	5632	2	RHENIUM CHLORIDES
1119	55	RADIUM 226	5591	2	RC-1	2934	9	RELIABILITY	7492	1	RHENIUM COMPLEXES
[illegible]	13	RADIUM 228	7472	1	RCROSS-CODE	1155	61	REMOTE CONTROL	4202	4	RHENIUM COMPOUNDS
904	[illegible]	RADON	1803	25	RE-ENTRY	542	195	REMOTE HANDLING	4752	3	RHENIUM FLUORIDES
5586	2	RADON ISOTOPES	5592	2	REACTION HEAT	7482	1	REMOVAL	4753	3	RHENIUM ISOTOPES
7448	1	RADON 202	66	1157	REACTION KINETICS	7483	1	RENAL CLEARANCE	7493	1	RHENIUM NITRIDES
7449	1	RADON 203	5593	2	REACTION MECHANISM	4199	4	RENE 41	[illegible]	15	RHENIUM OXIDES
7450	1	RADON 204	162	624	REACTIVITY	7484	1	RENE 62	7494	1	RHENIUM 175
7451	1	RADON 205	2360	14	REACTIVITY WORTHS	1530	35	RENORMALIZATION	7495	1	RHENIUM [illegible]
7452	1	RADON 206	164	622	REACTOR CORE	4200	4	REPAIR	7496	1	RHENIUM 179
7453	1	RADON 207	1678	29	REACTOR COUPLING	421	261	REPROCESSING	[illegible]	2	RHENIUM 180
7454	1	RADON 208	4749	3	REACTOR KINETICS	7485	1	REPRODUCIBILITY	7497	1	RHENIUM 181
7455	1	RADON 210	7473	1	REACTOR MATERIALS	431	255	REPRODUCTION	[illegible]	2	RHENIUM 183
7456	1	RADON 211	3096	8	REACTOR NOISE	4750	3	REPTILES	[illegible]	1	RHENIUM 184
7457	1	RADON 212	5594	2	REACTOR OSCILLATOR	108	798	RESEARCH REACTORS	[illegible]	15	RHENIUM 185
4195	4	RADON 219	4197	4	REACTOR PERIOD	5595	2	RESERPINE	2797	10	RHENIUM 186

Figure A-3

Alphabetic Listing of Keywords - 46821 Documents

whenever found. Abstracts would have been preferred (see end of Chapter 3), but they were not available in machine-readable form.

There were no experiments performed on the small Entry file (henceforth call the title word file). Due to missing data and bad tape data only 47,002 out of 47,055 documents in the large file could be processed. In addition, after processing it was found that 60 documents (0.13%) had no significant title words. Therefore the final title word file contained 46,942 documents.

A.3.2 Semi-Automatic Indexing

The above document titles now had to be analyzed to obtain significant words which can be used as keywords. The first step is to break the titles up into individual words. The ten break characters used for this purpose were: (blank) .) + - / , (= and '. Two special cases had to be taken care of. The first occurs when possession is indicated such as in "COW'S MILK". It is undesirable to break this up as "COW", "S", and "MILK" since "S" is the abbreviation of SULFUR. Therefore, an S following an ' (apostrophe) was ignored. The second special case is peculiar to this (and other similar) collection. Because of the inability of most computers to recognize subscripts and superscripts, chemical expressions such as the composition of water (H_2O) and a strontium isotope (^{90}Sr)

are represented as H/SUB 2/O and /SUP 90/SR respectively. At times this representation gets quite complex (i.e., /SUP 238/PUO/SUB 2/ or even 3D/SUP 10/4P/SUP 2/P/SUB /SUP 3///SUB 2//). The title partitioning program was designed to recognize these circumstances (by recognizing "/SUB" or "/SUP") and to consider these expressions as single words by ignoring any break characters (such as blanks) between slashes.

Table A-1 presents the steps involved in processing the title word file including the total number of terms (words or word stems) and the number of discrete terms found at each stage of processing. A number of these steps were combined in the actual processing but, for simplicity, are shown separately in the table.

A stop list was formed to eliminate some of the common words which could not be used as keywords. This list was obtained by considering other available stop lists [112] and by noting the most common words in the first 400 documents of this collection. A comparison of the twelve most often occurring common words with the top twelve words found in a recent analysis of 1,014,232 words of running text (broad cross-section of subjects) is shown in Table A-2. It should be noted that the top 12 out of 189 stop words accounted for 79 percent of the deleted title words.

Function Performed (from top down)	Number of discrete terms	Total number of terms
Break titles into words	20744	431022
Delete words on stop list	- 189 20555	- 135533 295489
Eliminate duplicate words for each document		- 3560 291929
Reduce terms by stem analysis	- 3423 17132	
Eliminate duplicate stems for each document		- 210 291719
Delete stems on new stop list	- 1755 15377	- 14380 277339
Reduce stems by simulated synonym dictionary	- 2068	
Eliminate duplicate stems for each document		- 198
Final Totals	13309	277141

Table A-1

Steps Involved in Processing Title Word File

	Title Words			Kucera and Francis [7(]	
Rank	Frequency	Percent of Total Words	Word	Percent of Total Words	Word
1	36009	8.3	OF	6.9	THE
2	15890	3.7	IN	3.6	OF
3	13113	3.0	AND	2.9	AND
4	11349	2.6	THE	2.6	TO
5	6396	1.5	ON	2.3	A
6	6364	1.5	FOR	2.1	IN
7	4560	1.1	A	1.0	THAT
8	3504	0.8	BY	1.0	IS
9	3132	0.7	WITH	1.0	WAS
10	2969	0.7	TO	0.9	HE
11	2202	0.5	FROM	0.9	FOR
12	1849	0.4	AT	0.9	IT
	107337	24.8		26.1	

Table [illegible]-2

Common Word Comparison

The third item of Table A-1 (it appears twice more in the table) eliminates duplicate words for each document. For example, if the last sentence of the previous paragraph was processed, one occurrence of the word "words" would have been deleted (all occurrences of "the" and "of" would have been deleted by the stop list).

In order to normalize the vocabulary to some extent, a stem analysis program was written. This program, a modified version of one used in the General Inquirer [131, 132], removes a number of different suffixes. Project SMART uses a table look-up procedure which, while more effective, seems to take considerably more computer time [27,81,119]. A flowchart of this program is shown in Figure A-4. The operation of this program removed enough suffixes to cause a reduction of 3423 in the number of discrete title words. Suffixes removed by this program are: s, e, es, ed, ing, ings, ion, ions, ly, edly, ingly, plus a doubled letter immediately followed by ed or ing. In addition, ies, ied and ily are replaced by the single letter y. However, in order to prevent the shortening or complete disappearance of short words (i.e., ion, gas, bee, wing, etc.), word length is not reduced below three letters. As a example of suffix removal, consider the following actual title words (frequency of occurrence are parenthesized):

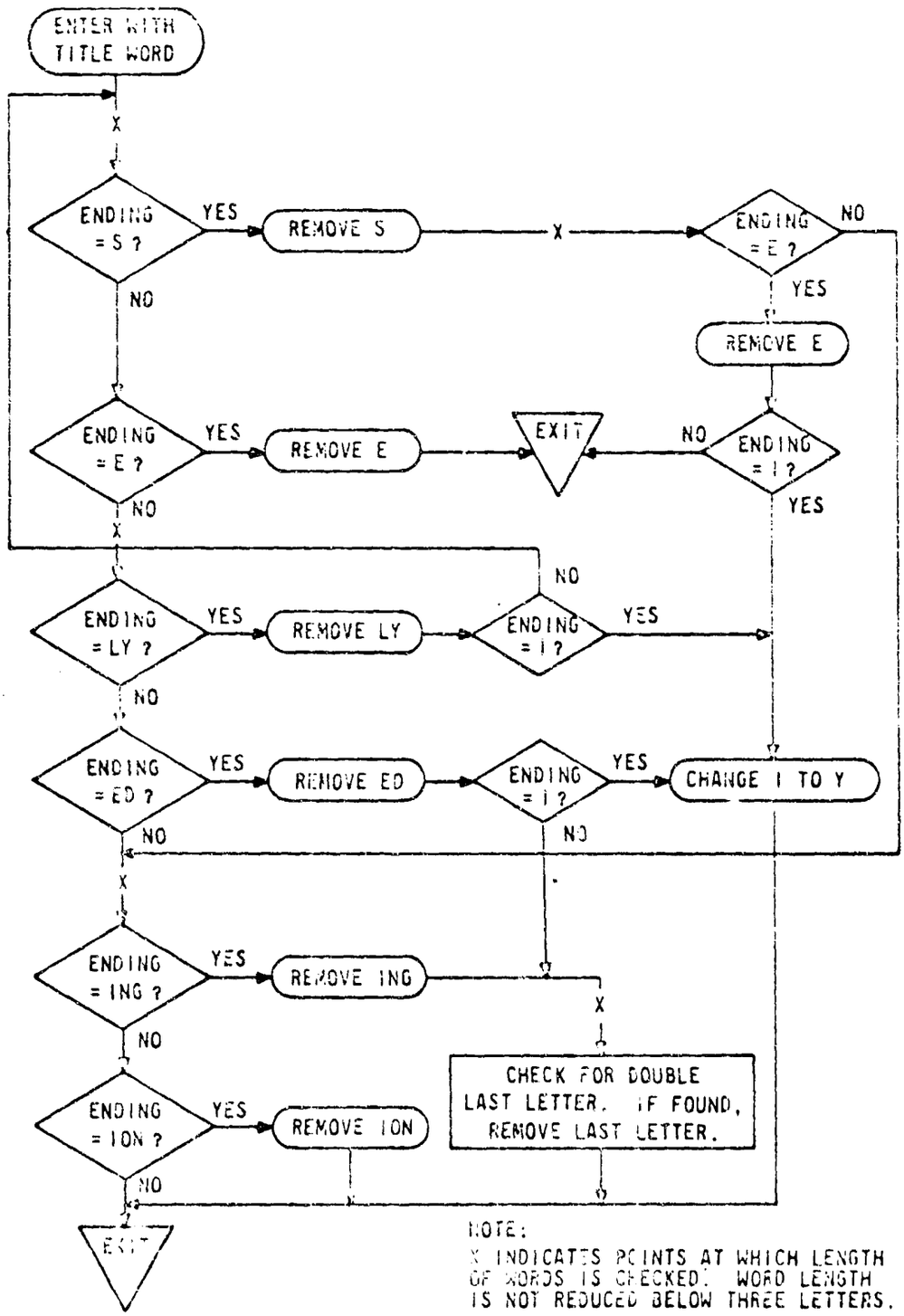


FIGURE A-4
SUFFIX DELETION ROUTINE

ESTIMATE (11)

ESTIMATED (5)

ESTIMATES (19)

ESTIMATING (27)

ESTIMATION (58)

ESTIMATIONS (1).

These words were all reduced to the single stem, ESTIMAT (121).

The final steps in the process of obtaining keyword stems were aided by manual intervention. A new stop list was formed and a synonym dictionary was simulated after inspection of the list of 17132 stems (see Table A-1). In an actual operating system, this need be done only once with few additions as the collection grows. About 1000 of the 1755 items on the new stop list were numbers. Many of the synonyms involved suffix deletions or changes and could have been incorporated in a more complete suffix removal program (e.g., combination of electrolytical with electrolytic). Others involved spelling errors and combination of British and American forms of the same words. Still others involved the combination of abbreviated forms and non-abbreviated forms of the same word or combination of chemical and English terms (e.g., converting H/SUB 2/O and 2H/SUB 2/O to WATER).

Once the keyword stems were obtained, they were converted to numbers in the same manner as was done for the

keywords of the keyword file (Fig. A-1). The 200 highest occurring word stems are shown in Figure A-5 (compare with those of the keyword file, Figure A-2). Full statistics for the title word file can be found in the next section.

A.4 File Statistics

Pertinent statistics for the three files studied are presented in Table A-3. One is reminded that the documents of the large keyword file and the title word file are essentially the same, both belonging to a collection of 47,055 documents. It should be noted that, as expected, the proportion of keywords with a single occurrence decreased from the small to the large keyword file. Also as expected, the proportion of unique keywords was highest for the title word file.

In Figure A-6 keyword frequency is plotted against rank (i.e., keyword order number). Zipf [155] found that when this curve was plotted for words of running text, a straight line resulted (Zipf's Law). However, as can be seen in Figure A-6, this does not hold for document index words. Houston and Wall [60] found, however, that when term frequency was plotted on logarithmic probability paper, a linear relationship was found to exist up to about the 95th percentile. The fact that this log-normal relationship holds for the files under consideration is shown in Figure A-7.

NUMB	FREQ	TERM	NUMB	FREQ	TERM	NUMB	FREQ	TERM	NUMB	FREQ	TERM
1	3082	RADIAT	51	775	ALLOY	101	422	FLOW	151	345	COMPOUND
2	3045	EFFECT	52	765	CALCULAT	102	471	TYP	152	343	SPECTRUM
3	2578	NUCLEAR	53	755	MATERIAL	103	469	CELL	153	342	FACTOR
4	2494	IRRADIAT	54	749	NUCLEI	104	466	DOS	154	340	SYMMETRY
5	2381	NEUTRON	55	741	CROS	105	462	PHYSIC	155	339	EMISS
6	2270	REACTOR	56	737	FAST	106	460	TRANSFER	156	338	POTENTIAL
7	2080	ELECTRON	57	720	SOLUT	107	459	SOLID	157	334	SPECTROSCOPY
8	1933	ENERGY	58	715	ELEMENT	108	456	EQUAT	158	333	CORRELAT
9	1831	STUDY	59	703	APPLICAT	109	450	DENSITY	159	330	COMPUTER
10	1801	MEASUREMENT	60	701	ATOM	110	448	SOLAR	160	329	ATMOSPHER
11	1759	HIGH	61	698	THERMAL	111	443	ACTIVAT	161	329	IMPROVEMENT
12	1672	PLASMA	62	692	PROCESS	112	440	SPAC	162	327	COLLIS
13	1593	SCATTER	63	689	CARBON	113	440	SUPERCONDUCT	163	327	NUCLEON
14	1595	SYSTEM	64	683	SOURC	114	434	PLASTIC	164	326	CHAMBER
15	1574	GAMMA	65	672	SECT	115	434	PLUTONIUM	165	324	NIOBIUM
16	1467	FIELD	66	667	INDUC	116	433	FUNCT	166	320	COMPLEX
17	1464	REACT	67	669	BEAM	117	432	BODY	167	318	OPERAT
18	1443	RAY	68	657	CURRENT	118	431	MESON	168	317	LIGHT
19	1367	PARTICL	69	657	HEAT	119	419	YIELD	169	317	SODIUM
20	1311	ANALYSI	70	655	U	120	414	SURFAC	170	314	DYNAMIC
21	1271	PRODUCT	71	634	ISOTOP	121	411	BEHAVIOR	171	312	SPECTROMETER
22	1233	METHOD	72	629	SPECTRA	122	410	ACID	172	309	EVALUAT
23	1221	X	73	619	P	123	409	ABSORPT	173	308	GROUP
24	1214	MAGNETIC	74	618	DETECTOR	124	403	ACCELERATOR	174	308	STABILITY
25	1136	STAT	75	617	LIQUID	125	403	COSMOLOGY	175	308	VARIAT
26	1154	THEORY	76	602	CHARG	126	402	O	176	307	LARG
27	1147	USE	77	600	RELAT	127	401	SPIN	177	305	FREQUENCY
28	1136	DETERMINAT	78	566	CHEMISTRY	128	399	HELIUM	178	304	PREPAR
29	1113	FUEL	79	566	FISS	129	393	DEPENDENCY	179	303	CRITICAL
30	1086	TEMPERATUR	80	563	EXCITAT	130	390	DATA	180	303	MECHANISM
31	1081	ION	81	560	ALPHA	131	388	DESIGN	181	301	KINETIC
32	1002	U	82	549	CRYSTAL	132	387	TREATMENT	182	297	HEAVY
33	975	EXPERIMENT	83	535	LEVEL	133	386	CHARACTERISTIC	183	296	LIN
34	971	RADIOACTIV	84	530	PLANT	134	386	TRANSIT	184	292	GENERATOR
35	937	WATER	85	525	MASS	135	384	DIFFUS	185	289	RELATIVITY
36	893	LOW	86	522	DEUTERIUM	136	381	BETA	186	289	TUR
37	884	METAL	87	521	CONTROL	137	380	SEPARAT	187	288	PROTECT
38	881	MODEL	88	521	PROBLEM	138	378	TECHNIQU	188	287	FORMAT
39	875	GAS	89	519	PROGRAM	139	371	PION	189	286	ENGINEER
40	867	INTERACT	90	508	PHAS	140	365	REG	190	284	S
41	867	STRUCTUR	91	508	TEST	141	363	ZIRCONIUM	191	283	[illegible]
42	863	POWER	92	502	POLARIZ	142	362	ACTIVITY	192	280	TRANSPORT
43	821	WAV	93	495	INFLUENC	143	360	TRITIUM	193	279	CONDUCT
44	815	PROTON	94	492	PULS	144	359	TUNGSTEN	194	278	RATIO
45	805	MEV	95	492	TIM	145	357	SINGL	195	277	CAPTUR
46	798	PROPERTY	96	491	PRESSUR	146	356	GEV	196	277	RAR
47	796	D AY	97	487	IONIZ	147	353	ELECTROMAGNETIC	197	276	DEVIC
48	792	DISTRIBUT	98	484	EXCHANG	148	351	FLUX	198	272	PARAMETER
49	787	RESONANC	99	479	ANGULAR	149	346	EARTH	199	272	STEEL
50	776	RAT	100	476	ELECTRIC	150	346	OXID	200	269	RANG

Figure A-5

200 Most Frequently Occurring Title Word Stems - 46942 Documents

	File		
	Small Keyword	Large Keyword	Title Word
Number of documents, N_d	2254	46821	46942
Number of discrete keywords, N_v	2557	8044	13309
Indexing method	manual with thesaurus	manual with thesaurus	semi-automatic on title
Total keyword occurrences	19,262	466,810	277,141
Average keywords per document, N_{kd}	8.54	9.96	5.90
Average documents per keyword (i.e., average number of keyword occurrences)	7.53	58.03	20.82
Number of unique keywords (i.e., occur only once)	992	2189	5879

Table A-3

File Statistics

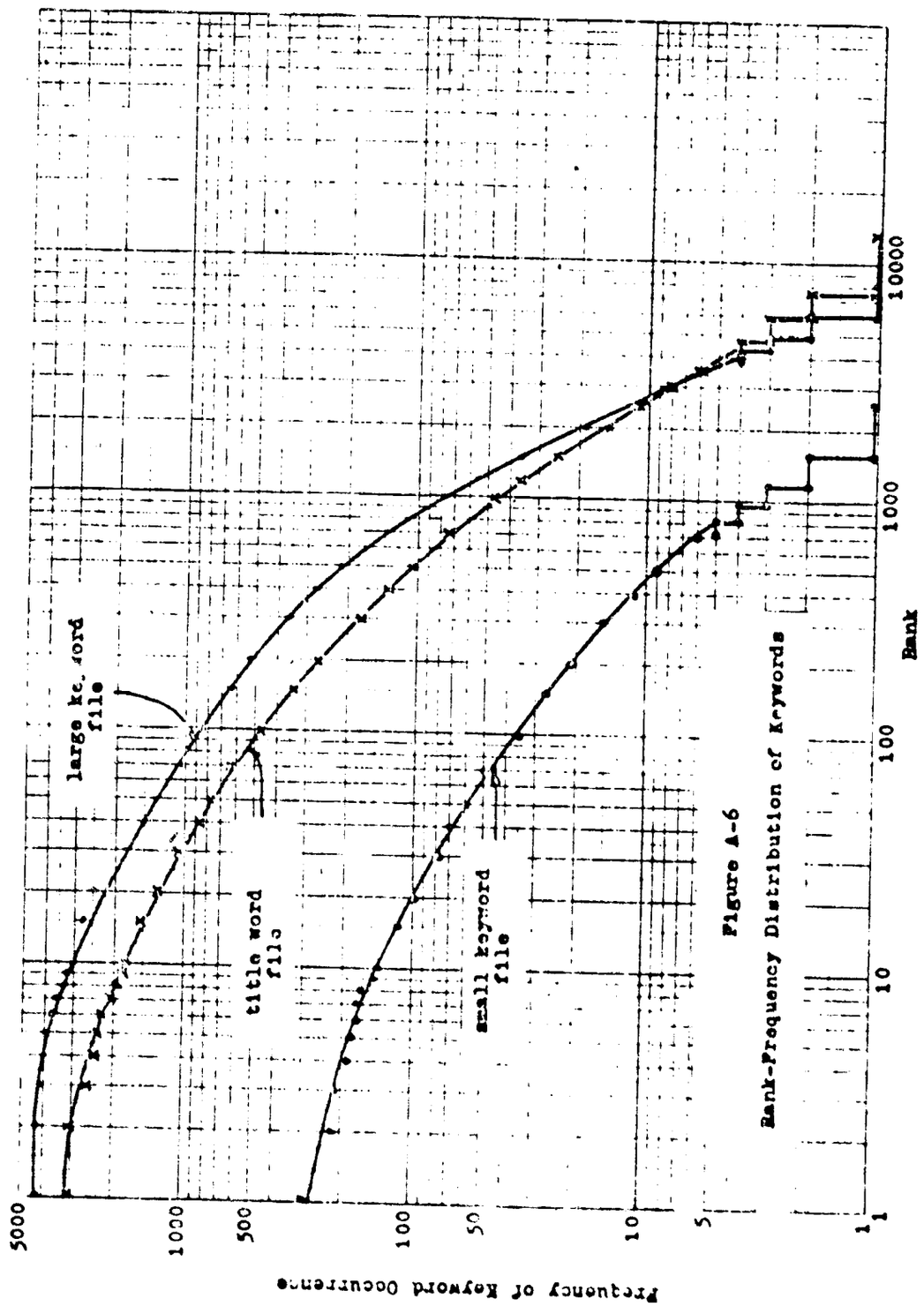


Figure A-6

Rank-Frequency Distribution of Keywords

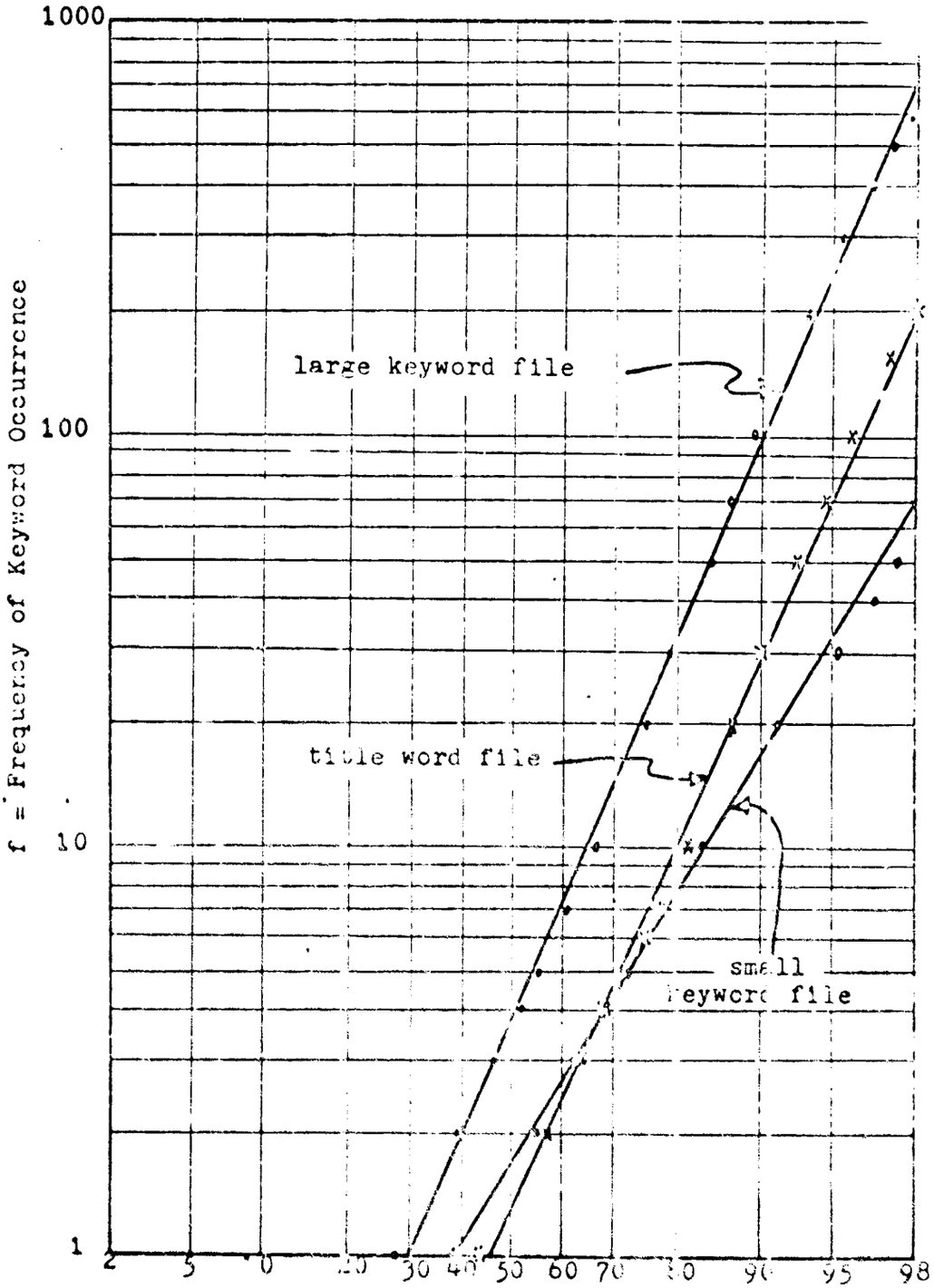


Figure A-7
Log-Probability Plot of Keyword Distribution

Based on ten systems, all of which follow this log-normal relationship, Houston and Wall went on to develop an expression relating vocabulary size to the total number of keyword occurrences. This formula is:

$$N_v = 3330 \log (K + 10000) - 12600$$

where the total number of keyword occurrences, $K = N_d \times N_{kd}$. Applying this formula to the files under study results in

	predicted	actual
small keyword	2250	2557
large keyword	6250	8044
title word	5550	13309.

The failure of the equation for the title word file is due to the fact that the equation was based on and seems to be only applicable to manual indexing systems which allow for vocabulary growth. A major reason for the actual vocabulary size being so much larger than the predicted size for the title word file is the large number of unique terms.

The distributions of the keywords per document for the three files are shown in Figure A-6.

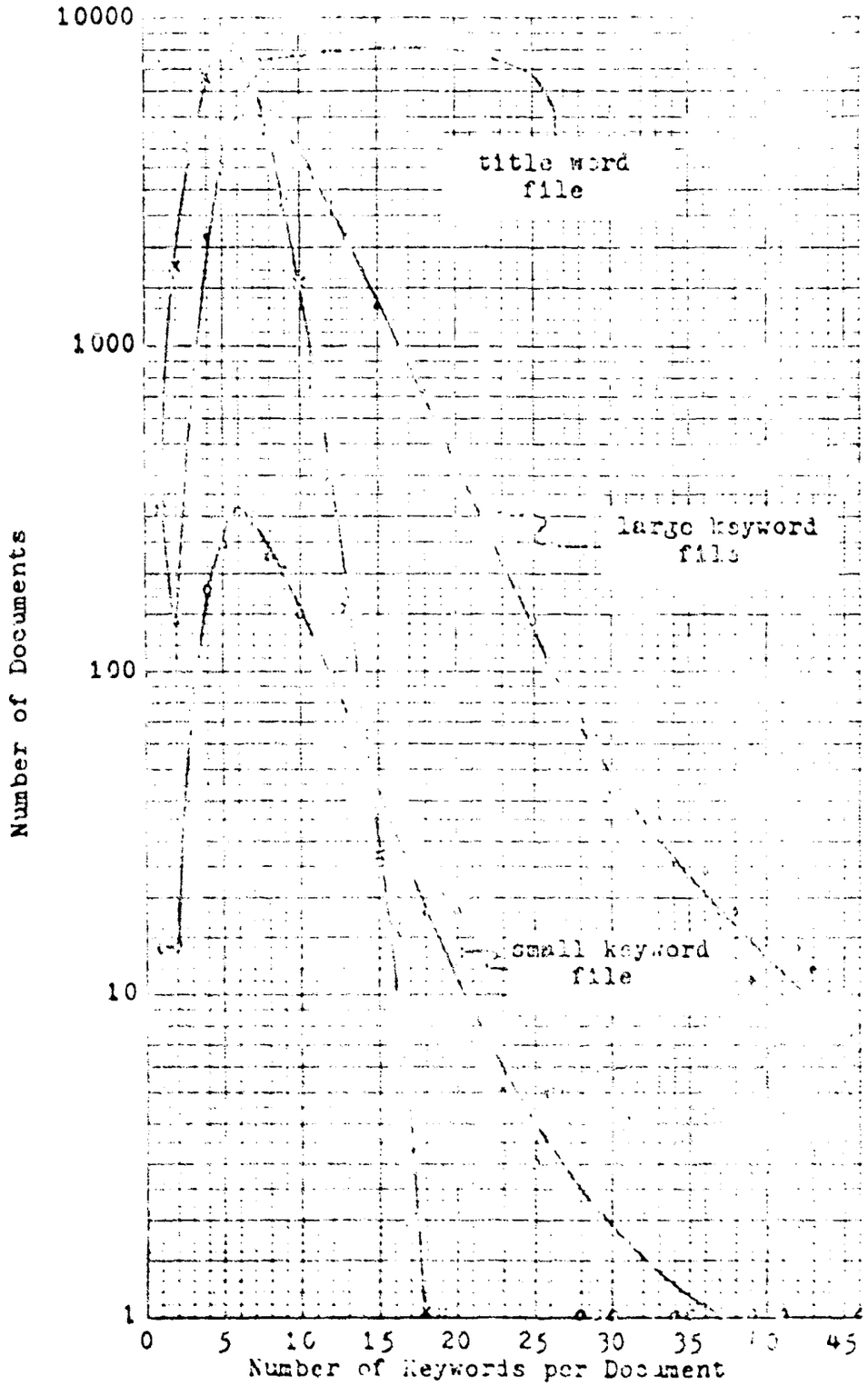


Figure A-8
Distribution of Number of Keywords per Document

APPENDIX B

DOCUMENT RETRIEVALS

B.1 Retrieval Requests

In order to test the ability of classification systems to group similar documents into cells (categories), actual retrievals based on real search requests were performed (see Chapter 5). These requests were obtained from Gloria Smith of the Lawrence Radiation Laboratories.

The requests originally consisted of conjunctions, disjunctions, and negations of EURATOM keywords and NSA categories. In all, there were 177 requests from 24 nuclear physicists or groups of nuclear physicists. These requests are in active use at Lawrence Radiation Laboratories, being serially matched against each semimonthly issue of the NSA Keyword File (see Appendix A). However, because of the expense of performing retrospective searches on serial files, these requests have not been used for retrospective searches.

Because these experiments were aimed at producing classification systems, the NSA categories could not be used for retrieval. In addition, the negations were also not used because (a) the types of statistics desired from the retrieval experiments would have been clouded by the use of negation and (b) the utility of negation would be lessened through the use of an on-line system which per-

mitted browsing. In order to eliminate the above request items while retaining the original meanings of the requests, some requests had to be altered somewhat and twelve of them dropped completely. This left 165 retrieval requests.

Since the requests were in terms of EURATOM keywords, no additional modifications were necessary to apply them to the keyword files. However, translation into word stems was required for the title word file. Where possible, this was done on a one-to-one basis; however, the aim was to maintain the meaning of the requests and not necessarily their exact forms.

Each request was of the form

$$(A_1 \text{ v } A_2 \text{ v} \ldots A_n) \text{ \& } (B_1 \text{ v } B_2 \text{ v} \ldots B_m) \text{ \& } (C_1 \text{ v } C_2 \text{ v} \ldots C_p)$$
$$\text{v } (D \text{ \& } E \text{ \& } F) \text{ v } (G \text{ \& } H \text{ \& } I) \text{ v } (J \text{ \& } K \text{ \& } L)$$

where v stands for logical OR, & stands for logical AND, A - L represent keywords, and n, m, and p are integers. The only essential part of this expressions is A_1. The integers n, m, and p can take on any values, but the highest encountered in the 165 requests was 39.

Some examples of typical requests are given in Figure B-1. Statistics for the 165 requests were tabulated and are presented in Table B-1. The terms used are defined in the following examples. A request of (A v B) & (C v D) & E has 4 (= 2 x 2 x 1) three conjunct conjunctions, 1 three conjunct expression, and 5 three conjunct tokens.

Request 1: (RADIATION INJURIES
OR RADIATION PROTECTION
OR RADIATIONS
OR RADIOACTIVITY)
AND (ANIMAL CELLS
OR BODY
OR BONES
OR MAN
OR PERSONNEL
OR TISSUES)
AND (GE... ...S
OR PHYSIOLOGY
OR STANDARDS
OR STATISTICS)

Request 11: PROJECT PLOWSHARE

Request 52: (HELIUM
OR SURFACE AREA
OR SURFACES
OR YTTRIUM)
AND (ABSORPTION
OR ADSORPTION
OR CLEANING
OR CRYOGENICS
OR LOW TEMPERATURE
OR MASS SPECTROMETERS
OR VACUUM)

OR (LIQUIDS AND NITROGEN)

OR (TRAPS AND NITROGEN)

OR (THERMIONICS AND LOW TEMPERATURE)

Request 139: COSMIC RADIATION
AND (MEASUREMENT
OR MODULATION)

Figure B-1

Typical Retrieval Requests

	Conjunctions	Expressions	Tokens
One conjunct	190	60	190
Two conjunct	1930	111	785
Three conjunct	2728*	22	269
Totals	4848	193	1244
Average per Question (+165)	29.4	1.2	7.5

*2280 of these are the result of just three expressions, the largest being 39 x 6 x 5 = 1170

Table B-1

Request Statistics

A request of (A v B) v (C & D) has 2 (=2) one conjunct conjunctions, 1 one conjunct expression, 2 one conjunct tokens, 1 (= 1 x 1) two conjunct conjunction, 1 two conjunct expression, and 2 two conjunct tokens.

B.2 Documents Retrieved

These requests were applied to the three files numerous times during the course of the experiments. Table B-2 shows the number of retrievals (the document abstract numbers were actually retrieved) and the number of documents they represent for each of the files. Because of the different indexes in the two large files, the number of documents retrieved was substantially different even though the files consist of essentially the same documents.

The discrepancies between the number of retrievals and the actual documents retrieved is due to the fact that some documents were retrieved in response to more than one request. The distributions of the number of times each document was retrieved are shown in Figure B-2. It was found that these distributions approximate straight lines (semi-log paper). It should be noted that the most "popular" document (large keyword file) was retrieved in response to 22 requests.

	File		
	Small Keyword	Large Keyword	Title Word
Documents in File	2254	46821	46942
Total Retrievals, 165 Requests	862	27635	18753
Average Retrievals per Request	5.2	167.5	113.7
Actual Documents Retrieved	601	15852	11726

Table B-2

Documents Retrieved

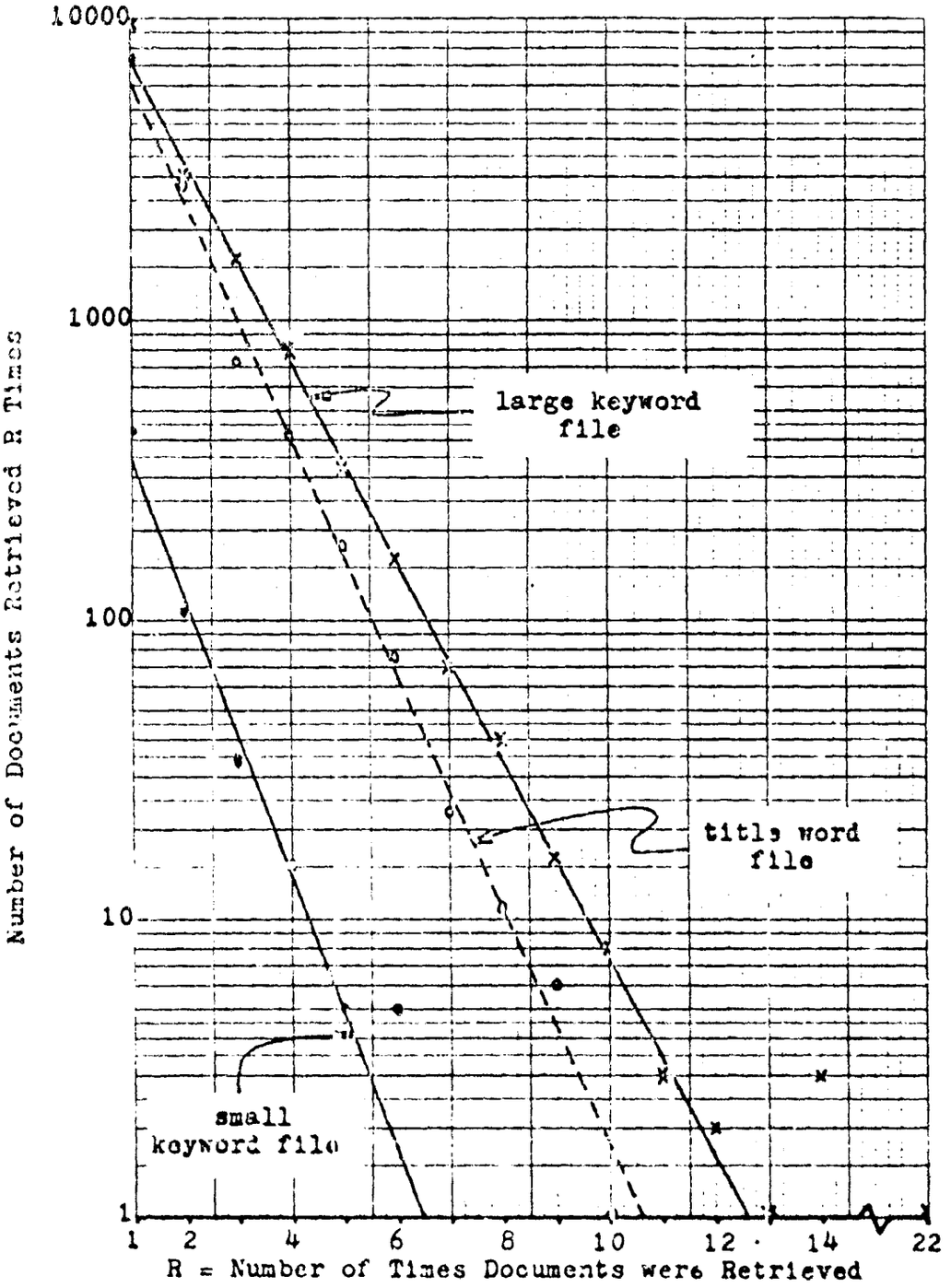


Figure B-2

Retrieved Documents Distributions